



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Despliegue de imágenes personalizadas del sistema operativo en un entorno corporativo

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Miguel Gómez Sabater

Tutor: Patricio Letelier Torres

2015-2016

Índice

1. Introducción	1
1.1. Antecedentes	1
1.2. Estado del arte	1
1.2.1. Análisis de la problemática	2
2. Objetivo	5
3. Soluciones/herramientas de software	7
3.1. Descripción y características de las herramientas	7
3.1.1 OpenGnSys	7
3.1.2. Servicios de implementación de Windows (WDS)	8
3.1.3. Symantec Ghost Solution Suite 3.1	9
3.1.4. Microsoft Deployment Toolkit (MDT)	11
3.1.5. Acronis Snap Deploy 5	13
3.1.6. System Center Configuration Manager (SCCM)	14
3.2. Comparativa de las herramientas	17
4. Microsoft Deployment Toolkit (MDT)	19
4.1. Descripción de la herramienta MDT	19
4.1.1. Conceptos previos	19
4.1.2. Tipos de despliegue. La elección de LTI, ZTI, o UDI	20
4.1.3. ¿Qué son las imágenes? ¿Qué es SysPrep?	22
4.1.4. Tamaño de las imágenes: pesadas, ligeras e híbridas	22
4.2. Requisitos de software para MDT	25
4.2.1. Sistemas operativos soportados	25
4.2.2. ADK de Windows 10	25
4.2.3. SQL 2014 Express Advanced	25
4.2.4. Software que hay que instalar	26
4.2.4.1. Orden de instalación del software	27
4.3. Componentes de MDT	29
4.3.1. Deployment Share (DS) o punto de despliegue	29
4.3.2. Rules o reglas	30
4.3.3. Boot images o imágenes de arranque	30



4.3.4. Operating Systems o sistemas operativos	30
4.3.5. Applications o aplicaciones	31
4.3.6. Out-of-Box Drivers o repositorio de controladores de dispositivos	31
4.3.7. Packages o paquetes	31
4.3.8. Task Sequences (TS) o secuencias de tareas	31
4.3.8.1. Plantillas de secuencias Task Sequence Templates	32
4.3.9. Selection Profiles o perfiles de selección	32
4.3.10. Logging o ficheros de registro	33
4.3.11. Monitoring o monitorización	33
4.4. Ejemplo práctico HELLO WORLD	33
4.4.1. Paso número 1: crear un Deployment Share	34
4.4.1.1. Carpetas que componen un DS	37
4.4.1.2. Actualizar el DS	38
4.4.2. Paso número 2: crear el usuario UsuarioMDT en el servidor MDT01	40
4.4.3. Paso número 3: importar un sistema operativo.	40
4.4.4. Paso número 4: crear una secuencia de tareas	44
4.4.5. Paso número 5: desplegar la imagen de pruebas	48
4.5. Crear una imagen de referencia	55
4.5.1. La plataforma de virtualización de Microsoft: Hyper-V	55
4.5.2. Instalar Hyper-V en Windows Server 2012 R2	56
4.5.3. Crear un ordenador de referencia en Hyper-V	57
4.5.3.1. Instalación del sistema operativo en el ordenador de referencia	59
4.5.3.2. Modo de auditoría	60
4.5.3.3. Instalar Office 2016	61
4.5.3.4. Actualizar el sistema operativo y Office 2016	61
4.5.3.5. Personalizar el escritorio	62
4.6. Capturar la imagen de referencia personalizada	63
4.6.1. Crear un DS para realizar capturas de imágenes	63
4.6.2. Importar un sistema operativo	64
4.6.3. Crear una secuencia de captura de la imagen de referencia	64
4.6.4. Ejecutar el fichero LiteTouch.vbs desde MDTCaptura	64

4.7. Despliegue de la imagen de referencia	66
4.7.1. Importar la imagen de referencia en el DS MDTLab	66
4.7.2. Crear secuencia de despliegue en el DS MDTLab	66
4.7.3. Desplegar la imagen de referencia	67
4.8. Añadir aplicaciones y controladores en el DS MDTLab	67
4.8.1. Añadir aplicaciones en el DS MDTLab	67
4.8.2. Añadir controladores de dispositivos en el DS MDTLab	70
4.9. Automatizando el despliegue. Pasar de utilizar LTI a utilizar ZTI	76
4.9.1. Rules (CustomSettings.ini) y BootStrap.ini	76
4.9.2. WMI y Windows Powershell	79
4.9.2.1. ¿Qué es WMI?	79
4.9.2.2. ¿Qué es Windows Powershell?	80
4.9.2.3. Utilizar Windows Powershell para obtener información del ordenador	80
4.9.2.3.1. Obtener el UUID y el número de serie del ordenador	81
4.9.2.3.2. Obtener la dirección MAC del adaptador de red	82
4.9.2.3.3. Obtener más información útil sobre el ordenador	83
4.9.3. Base de datos de MDT en SQL Server	84
4.9.3.1. Crear la base de datos	85
4.9.3.2. Permitir que el usuario UsuarioMDT pueda leer la base de datos de MDT	85
4.9.3.3. Almacenar información de los ordenadores en la base de datos	87
4.9.3.3.1. Crear un nuevo registro de un ordenador	87
4.9.3.3.2. Añadir información adicional para hacer despliegues ZTI	88
4.9.3.4. Añadir aplicaciones para instalar en ordenadores concretos	89
4.9.4. Utilizar WDS para arrancar el ordenador utilizando PXE	91
4.9.4.1. Agregar el servicio WDS en Windows Server 2012	91
4.9.4.2. Agregar imágenes de arranque de MDT	92
4.9.4.3. Seleccionar la imagen de arranque PXE por defecto	93
4.9.4.4. Activar entorno PXE en WDS	94
4.9.5. Como iniciar un despliegue	95
4.9.5.1. Iniciar despliegue utilizando un medio físico	96
4.9.5.2. Iniciar despliegue utilizando PXE	96
4.9.5.3. Iniciar despliegue ejecutando el fichero LiteTouch.vbs desde el DS MDTLab	96



4.9.6. Conseguir despliegue automatizado	97
4.9.6.1. Crear una tarea programada en el ordenador de referencia	97
4.9.6.2. <i>Script</i> para realizar despliegues totalmente automáticos	98
4.9.6.3. Crear tarea programada en el servidor MDT01	109
4.9.6.4. Ejecutar la tarea en el servidor MDT01	110
5. Conclusiones	113
6. Trabajos futuros	115
7. Bibliografía	116
Abreviaturas y siglas	119
Anexos	I
Anexo 1. El BUG de Windows 10	I

Listado de tablas

Tabla 1: Comparativa de las características de las seis herramientas.	17
Tabla 2: Comparativa de los tres tipos de despliegues: LTI, ZTI y UDI.	20
Tabla 3: Ventajas y desventajas de los despliegues totalmente automatizados y parcialmente automatizados.	21
Tabla 4: Ventajas y desventajas del tamaño de las imágenes.	23
Tabla 5: Resumen comparativo de las ediciones de Microsoft SQL Server.	26



Listado de figuras

Figura 1: Consola WEB de OpenGnSys.	8
Figura 2: Consola de los servicios de implementación de Windows.	9
Figura 3: Consola de Ghost Solution Suite.	11
Figura 4: Consola de MDT, el Deployment Workbench.	12
Figura 5: Consola de Acronis Snap Deploy 5.	14
Figura 6: La consola de SCCM.	16
Figura 7: Pantalla de instalación de Windows ADK 1511.	27
Figura 8: Consola de Microsoft SQL Server Management Studio.	28
Figura 9: El servicio SQL Server Browser en la consola de administración de equipos.	29
Figura 10: Entorno de MDT, el Deployment Workbench.	29
Figura 11: Como crear un nuevo Deployment Share.	34
Figura 12: La carpeta donde se creará el Deployment Share.	35
Figura 13: El nombre del recurso compartido asociado al Deployment Share.	35
Figura 14: Descripción ampliada del Deployment Share.	35
Figura 15: Las opciones de creación del Deployment Share.	36
Figura 16: Resumen de la creación del Deployment Share.	36
Figura 17: Mensaje de confirmación de la creación correcta del Deployment Share.	36
Figura 18: El Deployment Share MDTLab ya creado en el Deployment Workbench.	37
Figura 19: Actualización del Deployment Share MDTLab.	38
Figura 20: Opciones de actualización del Deployment Share MDTLab.	39
Figura 21: Resumen de la actualización del Deployment Share MDTLab.	39
Figura 22: Progreso de la actualización del Deployment Share MDTLab.	39
Figura 23: Informe de actualización correcta del Deployment Share MDTLab.	40
Figura 24: Como importar un nuevo sistema operativo al DS MDTLab.	41
Figura 25: Opciones de Importación de un nuevo sistema operativo.	41
Figura 26: Explorador de archivos para importar un nuevo sistema operativo.	42
Figura 27: Ruta completa al fichero install.wim para importar.	42
Figura 28: Opción de que no son necesarios los ficheros de instalación.	43
Figura 29: Nombre de la carpeta donde se almacenará el sistema operativo.	43
Figura 30: Resumen de la configuración para importar un sistema operativo.	43

Figura 31: Mensaje de comprobación de la correcta importación del sistema operativo.	44
Figura 32: El sistema operativo importado en el Deployment Workbench.	44
Figura 33: Como crear una nueva secuencia de tareas.	44
Figura 34: Datos identificativos de la nueva secuencia de tareas.	45
Figura 35: Selección de la plantilla a utilizar en el despliegue.	45
Figura 36: Se muestra el sistema operativo a desplegar.	46
Figura 37: Opciones del tipo de activación del sistema operativo.	46
Figura 38: Información relativa al propietario del sistema operativo	47
Figura 39: Opciones donde se puede introducir, o no, la contraseña del usuario Administrador.	47
Figura 40: Resumen de la creación de la tarea de secuencias.	48
Figura 41: Informe una vez finalizado el proceso de creación de la tarea de secuencias.	48
Figura 42: La tarea recién creada en el Deployment Workbench.	48
Figura 43: Asistente de despliegue LTI.	49
Figura 44: Especificación del servidor y el punto despliegue.	50
Figura 45: Credenciales para conectar con el punto despliegue.	50
Figura 46: Listado de secuencias de tareas a desplegar.	51
Figura 47: Datos referentes al ordenador.	51
Figura 48: Selección de no mover los datos del usuario.	52
Figura 49: Selección de no restaurar los datos del usuario.	52
Figura 50: Configuración local de idioma y hora.	53
Figura 51: Resumen de la configuración del despliegue.	53
Figura 52: Recopilar información para el despliegue.	54
Figura 53: Barra de progreso del despliegue de la imagen.	54
Figura 54: Mensaje de que el sistema operativo ha sido desplegado sin errores.	55
Figura 55: El administrador de servidor de Windows Server 2012 R2.	56
Figura 56: La creación de un conmutador virtual para Hyper-V.	57
Figura 57: Como crear la máquina virtual de referencia.	58
Figura 58: Máquina virtual de referencia creada en el administrador de Hyper-V.	59
Figura 59: Asistente para la configuración inicial del sistema operativo.	60



Figura 60: El sistema operativo en el modo de auditoria.	61
Figura 61: Actualización de Windows y Office 2016 desde WindowsUpdate.	62
Figura 62: Escritorio personalizado para todos los usuarios en Windows 10.	63
Figura 63: Mensaje de confirmación de la correcta captura de la imagen de referencia.	65
Figura 64: Explorador de archivos que muestra el fichero WIM capturado del ordenador de referencia.	66
Figura 65: Selección de la aplicación antivirus Kaspersky.	69
Figura 66: Instalación desatendida del antivirus Kaspersky al final del despliegue.	70
Figura 67: Los controladores importados en el Deployment Workbench.	71
Figura 68: Creación del perfil de selección de Windows 8.1 x64.	72
Figura 69: Configuración de la acción de inyectar controladores para que utilice el perfil de selección de Windows 8.1 x64.	73
Figura 70: Creación de la estructura de las carpetas de controladores en el Deployment Workbench.	75
Figura 71: Añadir la acción de establecer la variable DriverGroup001 de la secuencia de tareas con el valor: Windows 8.1 x 64 \% Model%.	75
Figura 72: Configurar la acción de inyectar controladores a el perfil de selección <i>Nothing</i> .	76
Figura 73: Ejecución del comando <code>Get-WmiObject -Class Win32_ComputerSystemProduct.</code>	82
Figura 74: Ejecución del comando <code>Get-WmiObject -Class Win32_ComputerSystemProduct Select-Object -Property UUID.</code>	82
Figura 75: Ejecución del comando <code>Get-WmiObject Win32_NetworkAdapterConfiguration.</code>	83
Figura 76: Ejecución del comando <code>Get-WmiObject Win32_NetworkAdapterConfiguration Select Description, Macaddress.</code>	83
Figura 77: Ejecución del comando <code>Get-WmiObject -Class Win32_BaseBoard.</code>	84
Figura 78: La base de datos MDT creada en el Deployment Workbench.	85
Figura 79: La carpeta de seguridad de la base de datos de MDT.	86
Figura 80: El nuevo inicio de sesión y permisos del UsuarioMDT en la BD de MDT.	87
Figura 81: El nuevo ordenador creado en el Deployment Workbench.	88
Figura 82: La consola de servicios de implementación de Windows.	92

Figura 83: La imagen de arranque LiteTouchPE_x64.wim agregada a imágenes de arranque de WDS.	93
Figura 84: Las propiedades de arranque de WDS en el servidor MDT01.	94
Figura 85: Las propiedades de respuesta PXE de WDS en el servidor MDT01.	95
Figura A1: Estructura de carpetas de controladores para Windows 8.1 en el DS.	I
Figura A2: Asignación de la variable DriverGroup001 a Windows 8.1 x64\%Manufacturer%\%Model%.	II
Figura A3: Estructura de carpetas de controladores para Windows 8.1 y para Windows 10 en el DS.	III
Figura A4: Captura de pantalla de Windows 10 con los datos del fabricante y del modelo en blanco.	IV
Figura A5: Captura de pantalla de Windows 8.1 con los datos del fabricante y del modelo.	IV
Figura A6: Hilo creado en el foro de MDT en Microsoft Technet.	V
Figura A7: Carpeta Scripts con el fichero AliasUserExit.vbs.	VIII
Figura A8: Estructura de carpetas actual para los controladores de Windows 10.	XVI
Figura A9: Asignación de la variable DriverGroup001 a Windows 10 x64\%ModelAlias%.	XVII

Listado de códigos

Código 1: Contenido del fichero CUSTOMSETTINGS.INI del DS MDTLab.	77
Código 2: Contenido del fichero CUSTOMSETTINGS.INI para hacer ZTI.	78
Código 3: Contenido del fichero BOOTSTRAP.INI del DS MDTLab.	78
Código 4: Contenido del fichero BOOTSTRAP.INI para hacer ZTI.	79
Código 5: Rules para instalar aplicaciones en un ordenador específico.	91
Código 6: Líneas de Rules para no hacer copia de seguridad de los datos de usuario.	96
Código 7: Contenido del fichero DespliegueAutomatico.PS1.	109
Código 8: Contenido del fichero ORDENADORES_DESPLIEGUE.XML.	109
Código 9: Contenido del fichero DESPLIEGUE_MDT01.LOG.	111
Código A1. Líneas modificadas del fichero ZTIGather.WSF.	VII
Código A2: Contenido del fichero AliasUserExit.vbs.	XV
Código A3: Líneas modificadas del fichero CUSTOMSETTINGS.INI para utilizar el <i>script</i> AliasUserExit.	XV

1. Introducción

1.1. Antecedentes

En una universidad hay que mantener un gran número de ordenadores en funcionamiento, y todos ellos con el mismo software instalado, de modo que los usuarios siempre puedan disponer del software que necesitan indistintamente del aula donde se encuentren.

El problema es que no todos los ordenadores son iguales. Dentro de una misma aula de informática podemos encontrarnos con varios modelos de ordenador y con configuraciones de hardware muy distintas, como puede ser; distintos procesadores, tarjetas gráficas, adaptadores de red y monitores con distinta relación de aspecto.

Necesitamos disponer de alguna herramienta que permita realizar ese mantenimiento fácilmente y que requiera una mínima o ninguna interacción para llevarlo a cabo. Debe de ser capaz de suministrar los controladores de dispositivos que necesita cada modelo de ordenador, la resolución de la pantalla y otros datos relativos a su configuración.

En este TFG, analizaremos varias herramientas que permiten realizar este mantenimiento para poder seleccionar la aplicación que más nos interese. Para ello realizaremos pruebas de despliegue con ellas y finalmente, si es posible, automatizaremos al máximo el proceso de despliegues de las imágenes.

1.2. Estado del arte

Para poder trabajar con un ordenador, primero tenemos que instalar un sistema operativo, instalar las aplicaciones que necesitemos y posteriormente aplicar las configuraciones que precisemos. En el caso de un único ordenador, o de pocos ordenadores, podemos realizar la tarea manualmente (uno por uno) pero cuando el número de ordenadores es elevado, ya no resulta viable el hacerlo manualmente. El tiempo que tendríamos que invertir en completar la tarea sería cuantioso.

Los ordenadores se componen de hardware y software. Las definiciones de hardware y software de acuerdo a la página web de Wikipedia, son:

La palabra hardware se refiere a todas las partes físicas de un sistema informático; sus componentes son eléctricos, electrónicos, electromecánicos y mecánicos. Cables, gabinetes o cajas, periféricos de todo tipo y cualquier otro elemento físico involucrado [1].

Se conoce como software al equipo lógico o soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas [2].

Los componentes lógicos incluyen, entre muchos otros, el llamado software de sistema, tal como el sistema operativo, que básicamente permite al resto de los programas funcionar adecuadamente, facilitando también la interacción entre los componentes físicos y el resto de las aplicaciones, y proporcionando una interfaz con el usuario; las aplicaciones informáticas, tales como el procesador de texto, que permite al usuario realizar todas las tareas concernientes a la edición de textos.

El mantenimiento del hardware supone el cambio de algún componente físico, bien por fallo o por actualización del mismo, y normalmente no requiere el invertir una gran cantidad de tiempo en ello. Sin embargo, el software es una tecnología muy dinámica:

- Se incorporan mejoras en los sistemas operativos.
- Se publican versiones nuevas de los programas regularmente.
- Se detectan fallos de seguridad que son solucionados mediante parches de software.
- Hay actualizaciones de los antivirus prácticamente a diario.
- Los fabricantes de hardware lanzan nuevos controladores de dispositivos.

Para asegurar el correcto funcionamiento de los ordenadores hay que aplicar regularmente dichas actualizaciones, y algunas veces, la cantidad y tamaño de dichas mejoras y actualizaciones puede ser muy grande. Otro punto a tener en cuenta es que en las grandes corporaciones normalmente no se encuentran ubicadas en una única localización, sino que tienen varias sedes distribuidas a lo largo de un país, el mantenimiento de la infraestructura del software se complica aún más. Para solucionar el problema del mantenimiento de la infraestructura del software en muchos ordenadores simultáneamente, necesitamos disponer de alguna herramienta que nos facilite el aplicar fácilmente a un grupo de ordenadores todas esas mejoras y actualizaciones de forma centralizada y coordinada, además debería realizarse por medio de un proceso que requiera mínima o ninguna interacción con el usuario.

1.2.1. Análisis de la problemática

En la Escuela Técnica Superior de Ingeniería Agronómica y del Medio Natural de la Universitat Politècnica de València, nos encontramos con que cada vez se imparte más docencia en las aulas de informática, donde tenemos entre diez y cuarenta ordenadores en cada aula. Para optimizar el uso de las mismas, se mantienen todas con el mismo sistema operativo y con las mismas aplicaciones instaladas, de modo que, si por cualquier motivo no se puede utilizar un aula, se pueda utilizar otra distinta, sin que haya que preocuparse del software que tenga instalado.

La gestión de equipos informáticos incluye la instalación del sistema operativo, normalmente compuesto por un sistema de archivos junto a las aplicaciones de usuario, contenido dentro de una partición del disco duro del equipo. Esta instalación de sistemas operativos y mantenimiento se complica al aumentar el parque informático a gestionar, por lo que se requiere el uso de técnicas de clonación.

La clonación es el proceso de duplicar un sistema operativo instalado en un ordenador de referencia a uno o más ordenadores de destino. Las herramientas informáticas usadas para la clonación se clasifican dependiendo de su capacidad para acceder al sistema operativo en:

- Clonación basada en sectores: disco o partición.
- Clonación basada en el contenido de la partición.
- Clonación basada en el contenido del sistema de archivos.

La clonación basada en sectores duplica cada uno de los sectores del dispositivo. La característica principal de este tipo de clonación es la mayor duración del proceso y el requisito de tamaño mínimo en el dispositivo de destino: al menos el mismo número de sectores que el origen.

La clonación basada en el contenido de la partición tiene la ventaja de trabajar sólo con datos útiles -el espacio no usado por el usuario no es utilizado-, por lo tanto, la rapidez del proceso de clonado aumenta. Pero el requisito del tamaño mínimo del dispositivo de destino se mantiene: al menos el mismo tamaño que la partición de origen.

La clonación basada en el contenido del sistema de archivos ofrece la ventaja de copiar los datos de usuario únicamente y considerar el tamaño de los datos como el sistema de archivos a clonar -independiente de la partición que la incluye- con lo cual el tamaño de la partición destino debe ser al menos igual a los datos de usuario y no al tamaño de la partición de origen [3].

La flexibilidad de la operación de clonado de los ordenadores nos tiene que permitir el poder desplegar una imagen personalizada en el disco duro de múltiples ordenadores con configuraciones de hardware muy distintas, como puede ser:

- Distintos tipos de procesador (CPU).
- Diferentes tarjetas gráficas.
- Adaptadores de red de distintos fabricantes.
- Monitores con diferentes resoluciones y relaciones de aspecto.

La herramienta debe de ser capaz de almacenar datos importantes relativos a la configuración de cada ordenador, como son:

- El nombre del ordenador.
- El dominio o grupo de trabajo donde se agregará el ordenador.
- Los controladores de dispositivos.
- La configuración de la resolución de la tarjeta gráfica.
- Los parámetros del adaptador de red.

Conviene aclarar que no es posible coger un disco duro de un ordenador e instalarlo en otro ordenador y esperar que funcione. Hay que tener en cuenta los siguientes factores:

- El hardware del ordenador no tiene por qué ser igual.
- Dentro de la red local, el nombre del ordenador debe de ser distinto y único, pues sino, habría nombres duplicados en la red provocando conflictos de identificación del ordenador.
- Los controladores de dispositivos instalados en un ordenador no tienen por qué funcionar en otro distinto. El sistema operativo tiene que explorar el ordenador para determinar los controladores que necesita para cada dispositivo del sistema.
- En algunos casos, el sistema operativo tiene que verificar el hardware para comprobar la licencia.

La acción de replicar desde uno o varios ficheros almacenados en un servidor hacia múltiples ordenadores se conoce como despliegue (*Deploy*) y diremos que la herramienta desplegara una imagen hacia ellos.

2. Objetivo

El objetivo de este proyecto es seleccionar y aplicar una herramienta que nos permita realizar la operación de despliegue de imágenes personalizadas del sistema operativo y aplicaciones; con la mínima interacción del usuario, y si es posible, sin ninguna interacción.

Partiremos analizando las herramientas disponibles en el mercado para realizar la tarea de despliegue, analizando las mismas y donde las siguientes características serán las más valoradas:

- Coste económico de las licencias de utilización.
- Integración con los sistemas informáticos actuales en la ETSIAMN.
- Facilidad de uso.
- Flexibilidad.
- Posibilidad de automatización, interacción mínima o nula con el usuario o técnico.

De este análisis elegiremos la herramienta que más se adecue a nuestras necesidades.

3. Soluciones/herramientas de software

Vamos a ver seis herramientas que permiten realizar la clonación de varios ordenadores simultáneamente.

3.1. Descripción y características de las herramientas

Las herramientas que vamos a analizar y comparar son las siguientes:

- OpenGnSys
- Servicios de implementación de Windows (WDS)
- Symantec Ghost Solution Suite 3.1
- Microsoft Deployment Toolkit (MDT)
- Acronis Snap Deploy 5
- System Center Configuration Manager (SCCM)

3.1.1 OpenGnSys

Es una aplicación libre y abierta para la gestión y el despliegue de sistemas operativos [4].

El Proyecto OpenGnSys reúne el esfuerzo conjunto de varias Universidades Públicas Españolas, para disponer de una serie de herramientas libres y abiertas que constituyan un sistema completo, versátil e intuitivo, para la gestión y clonación de equipos. Este sistema permite la distribución, instalación y despliegue de distintos sistemas operativos.

La arquitectura de OpenGnSys es suficientemente flexible para adaptarse a las necesidades de los diferentes modelos de redes de ordenadores, tanto de empresas como de instituciones, pudiendo ser utilizado en distintos tipos de escenarios:

- Gestión centralizada de Unidades TIC de Apoyo a Docencia e Investigación.
- Mantenimiento del parque de ordenadores personales de los miembros de una Institución.
- Despliegue y mantenimiento de los servidores de un Centro de Datos en particular.
- Administración de repositorios de clonación de equipos para los Servicios de soporte (*helpdesk*) de una Institución.

Un sistema OpenGnSys está formado por dos tipos de servidor, según los servicios que presta [5]:

- Servidor de administración: único para cada sistema OpenGnSys, ofrece los servicios de administración a través de la consola web y ejecuta el demonio (*ogAdmServer*) que distribuye las tareas configuradas desde la consola web entre los diferentes servidores de repositorio.
- Servidor de repositorio: puede haber tantos como se desee. Ofrece los servicios de arranque por PXE, comparte por Samba los directorios necesarios para los clientes, almacena las imágenes de los discos de los clientes, recibe del servidor de administración las tareas que debe ejecutar los clientes y, en general, ofrece todos los servicios que utilizan directamente los clientes.
 - El equipo modelo para generar las imágenes debe ser de un hardware similar, y sobre todo en el asunto de la arquitectura.
 - Y en Windows, especialmente el hardware de la controladora del disco.

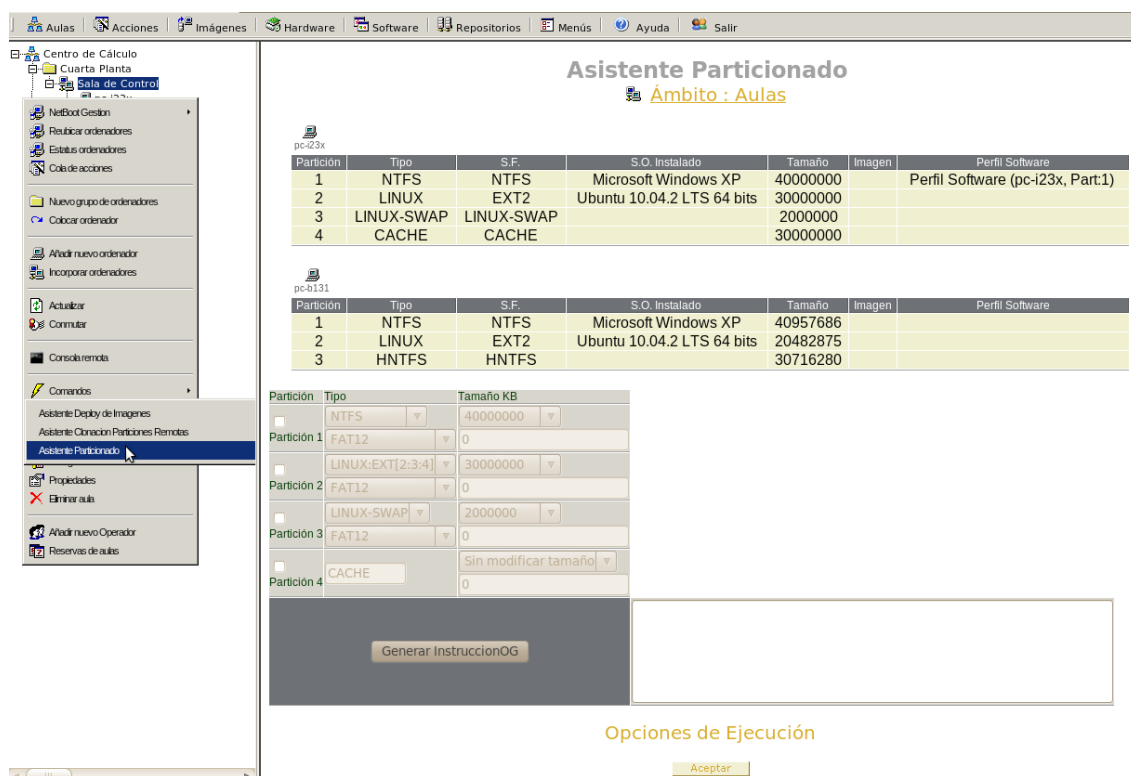


Figura 1: Consola WEB de OpenGnSys.

3.1.2. Servicios de implementación de Windows (WDS)

Servicios de implementación de Windows, Windows Deployment Services (WDS) permite implementar los sistemas operativos Windows a través de la red, lo que significa que no es necesario instalar cada sistema operativo directamente desde un CD o DVD.

Ventajas de utilizar los servicios de implementación de Windows: [6]

- Permite la instalación basada en la red de sistemas operativos Windows, lo que reduce la complejidad y el costo en comparación con las instalaciones manuales.

- Admite la implementación de imágenes para entornos mixtos, incluidos Windows 7 y Windows Server 2008 R2 hasta Windows 10 y Windows Server 2012 R2.
- Usa tecnologías de instalación de Windows estándar, incluidos el entorno de preinstalación de Windows (Windows PE), los archivos .wim y la instalación basada en imagen.
- Transmite datos e imágenes con la funcionalidad de multidifusión.
- Permite crear imágenes de un equipo de referencia mediante el asistente para capturar imágenes, una alternativa a la herramienta ImageX.
- Permite agregar paquetes de controladores al servidor y configurarlos para que se implementen en equipos cliente junto con la imagen de instalación.

WDS está desarrollado por Microsoft y solo despliega imágenes del sistema operativo Windows. Aunque es parte integrante de Windows Server 2012 y de Windows Server 2012 R2, por todas sus limitaciones, no se le puede considerar como una solución para el despliegue de imágenes. Sin embargo, es una pieza clave para otras herramientas de despliegue como Microsoft Deployment Toolkit (MDT). Aunque no sea una solución en sí mismo, se puede utilizar como soporte para poder arrancar los ordenadores utilizando el protocolo PXE, y si es necesario, realizar despliegue de imágenes utilizando multidifusión.

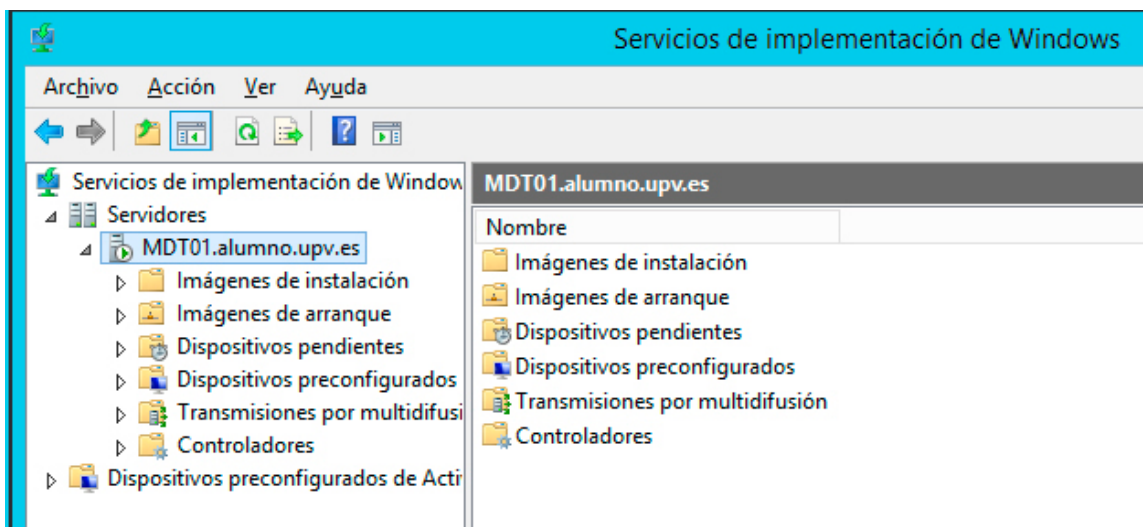


Figura 2: Consola de los servicios de implementación de Windows.

3.1.3. Symantec Ghost Solution Suite 3.1

La solución Symantec Ghost Suite esta considerada como líder en la industria para el despliegue de imágenes a ordenadores de escritorio, portátiles, tabletas y servidores. Desde una única consola de gestión, puede rápida y fácilmente migrar a la última versión del sistema operativo, hacer inventario de los ordenadores, implementar el software y realizar configuraciones personalizadas a través de múltiples plataformas de hardware y varios tipos de sistemas operativos, incluyendo Windows, Mac y Linux [7].

Reduce los costos de implementación, la complejidad y el tiempo. Por ejemplo, se puede construir un sistema de referencia mediante el asistente de instalación del sistema operativo sin supervisión, incluyendo la adición de software y la configuración requerida. Una vez que su sistema de referencia se ha construido de forma automática se puede tomar una imagen e implementar masivamente a otros sistemas de hardware, totalmente independiente y en cualquier ubicación remota del mundo. Como parte de la implementación el proceso puede agregar software específico dependiendo del sistema o de un grupo para aplicar una configuración específica, y personalizar totalmente las máquinas individualmente basadas en una serie de atributos.

Características:

- Es uno de los mejores sistemas para captura y despliegue de discos duros.
- Permite la instalación del sistema operativo mediante un *script*.
- Permite la migración de la configuración de usuario de y sus aplicaciones.
- Dispone de un sistema inteligente de distribución de controladores de dispositivo.
- Permite la ejecución de tareas remotamente y en secuencia.
- Soporta la implementación sobre ordenadores nuevos.
- Tiene soporte para los sistema operativos Windows, Mac y Linux.
- Presenta compatibilidad con algunas tabletas (modelos seleccionados).
- Permite un gran nivel de automatización del entorno.

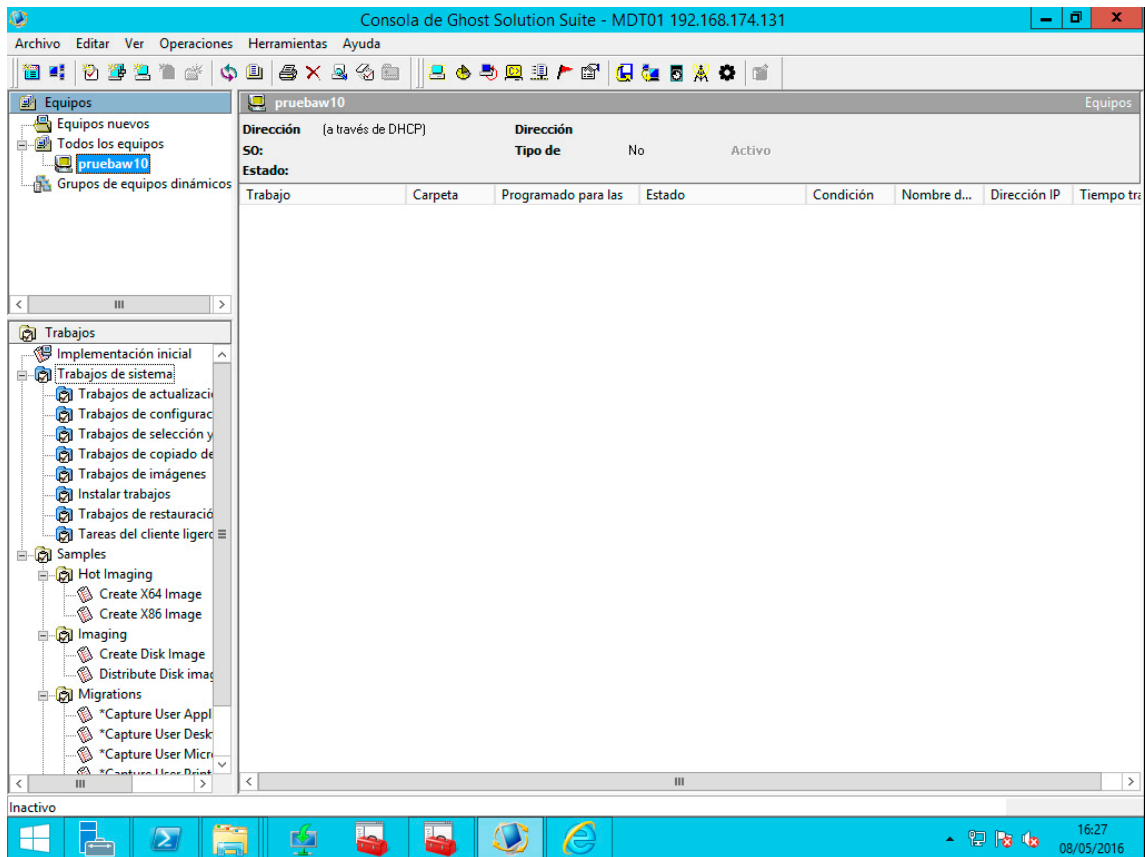


Figura 3: Consola de Ghost Solution Suite.

3.1.4. Microsoft Deployment Toolkit (MDT)

MDT es un producto hecho por Microsoft para ayudar en el despliegue del sistema operativo Windows. MDT proporciona una colección unificada de herramientas, procesos, y asistencia para automatizar el despliegue de imágenes en ordenadores de escritorio y servidores. Además de reducir el tiempo de despliegue y estandarizar las imágenes de los ordenadores, MDT ofrece opciones mejoradas de administración de seguridad y de configuración de los ordenadores [8].

Características: [9]

- Soporta los sistemas operativos cliente Windows 7, Windows 8, Windows 8.1 y Windows 10.
- Soporta los sistemas operativos Windows Server 2008 R2, Windows Server 2012, y Windows Server 2012 R2.
- Soporta el despliegue hacia ordenadores que usan Unified Extensible Firmware Interface (UEFI) versión 2.3.1.
- Soporta el despliegue hacia máquinas que requieren el nuevo formato GUID (globally unique identifier) partition table (GPT). Está relacionado con UEFI.
- Soporta la ejecución de *scripts* de PowerShell versión 3.0, además de la versión anterior PowerShell 2.0.

- Permite añadir múltiples cuentas de usuario al grupo local de administradores del ordenador de destino bien utilizando una configuración guardada previamente o utilizando un asistente.
- Proporciona plantillas de secuencias de tareas ya preparadas para la implementación de Windows en un archivo de disco duro virtual (VHD).
- Muestra información adicional del progreso del despliegue e incorpora una interfaz de usuario más limpia.
- Permite controlar el estado de los despliegues que se están ejecutando.
- Proporciona procedimientos de partición mejorados que aseguran el correcto despliegue sin importar la estructura del disco duro.
- Se puede activar durante la fase de Windows Preinstallation Environment (Windows PE), ahorrando tiempo en el proceso de encriptación.
- Soporta la ejecución de USMT (offline user-state migration), durante la fase de Windows PE del despliegue.
- Instala o desinstala roles y características de Windows como parte del despliegue.
- Se pueden integrar las herramientas de DaRT en la imagen de arranque.
- Soporte de aprovisionamiento de aplicaciones de Modern UI.
- Se pueden expandir las características propias de MDT añadiendo *scripts* personalizados, servicios web, System Center Orchestrator runbooks, *scripts* de PowerShell y VBScripts.

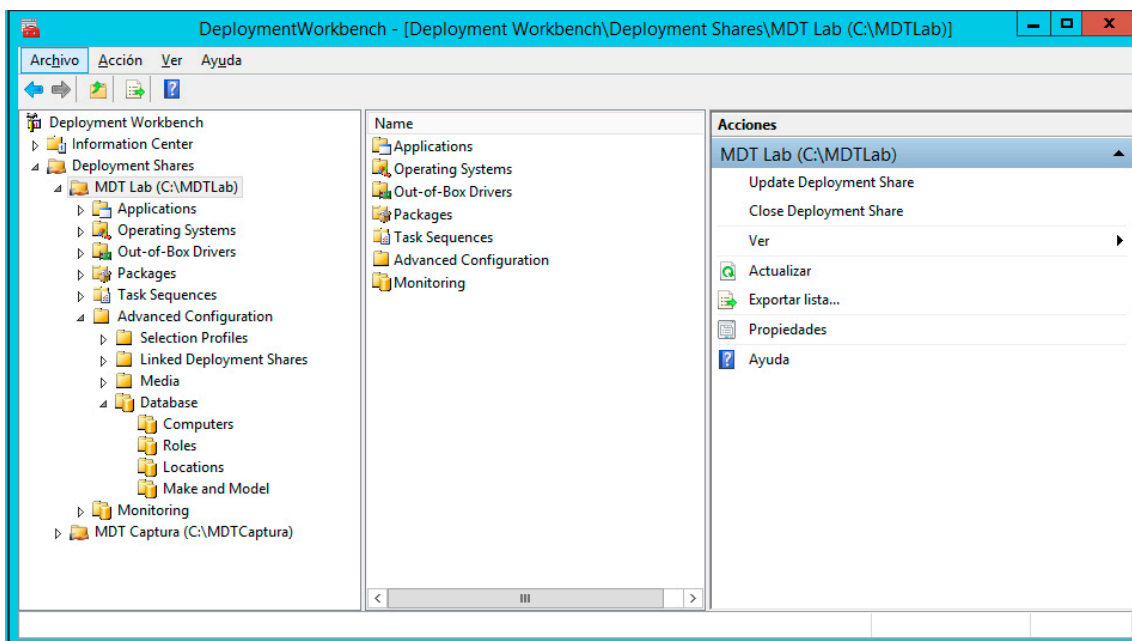


Figura 4: Consola de MDT, el Deployment Workbench.

3.1.5. Acronis Snap Deploy 5

Acronis Snap Deploy es una solución idónea para el despliegue rápido de inicio sin sistema operativo (*bare-metal*) y para un gran número de estaciones de trabajo y servidores, así como el despliegue sobre el mismo hardware en entornos de educación, formación y laboratorios. Acronis Snap Deploy permite aprovisionar cientos de estaciones de trabajo y servidores a la misma velocidad que si fuera uno solo. Está basado en Acronis AnyData Engine que permite crear una imagen de disco de cualquier configuración estándar, incluyendo el sistema operativo, la configuración, los ficheros y todas las aplicaciones, y, simultáneamente, desplegarla en múltiples equipos en un solo paso. Acronis AnyData Engine es el corazón de la suite que permite capturar, almacenar, recuperar, controlar y acceder a los datos. Acronis Snap Deploy incluye Acronis® Universal Deploy®, que permite desplegar la misma imagen a otros PCs y servidores incluso si son de distinto fabricante. Gestiona también configuraciones específicas de equipos, incluyendo el nombre, las direcciones IP, la pertenencia a dominio y otras configuraciones. Además, al finalizar el despliegue, Acronis Snap Deploy permite modificar de forma sencilla la configuración de Windows.

Características:

- Desde el punto central de administración se puede de forma rápida desplegar una imagen a múltiples ordenadores o servidores *vía unicast o multicast*.
- El interfaz simplificado de usuario ahorra tiempo y reduce la posibilidad de errores.
- Proporciona flexibilidad permitiendo desplegar Windows Server, ordenadores de sobremesa y tabletas basadas en Windows, y Linux.
- Se puede escoger entre cuatro opciones de despliegue: manual, automática, iniciada por el usuario y planificada. Es posible desplegar una imagen en modo independiente incluso si el equipo no está en la red.
- Acronis Universal Deploy admite el desplegar la misma imagen en un hardware distinto, simultáneamente.
- Se pueden asignar configuraciones individuales para cada ordenador o servidor para garantizar que reciben la configuración correcta. El progreso y el estado del despliegue está monitorizado.
- Es posible arrancar de forma automática los ordenadores para el despliegue con Wake-on-LAN, asegurando que se lleva a cabo acorde a la programación.
- Las opciones de licenciamiento pueden ser en función de: el número de despliegues, por número de equipos, y por sitio, para instalaciones más grandes.
- Soporta el despliegue de sistemas operativos Windows y Linux [10].

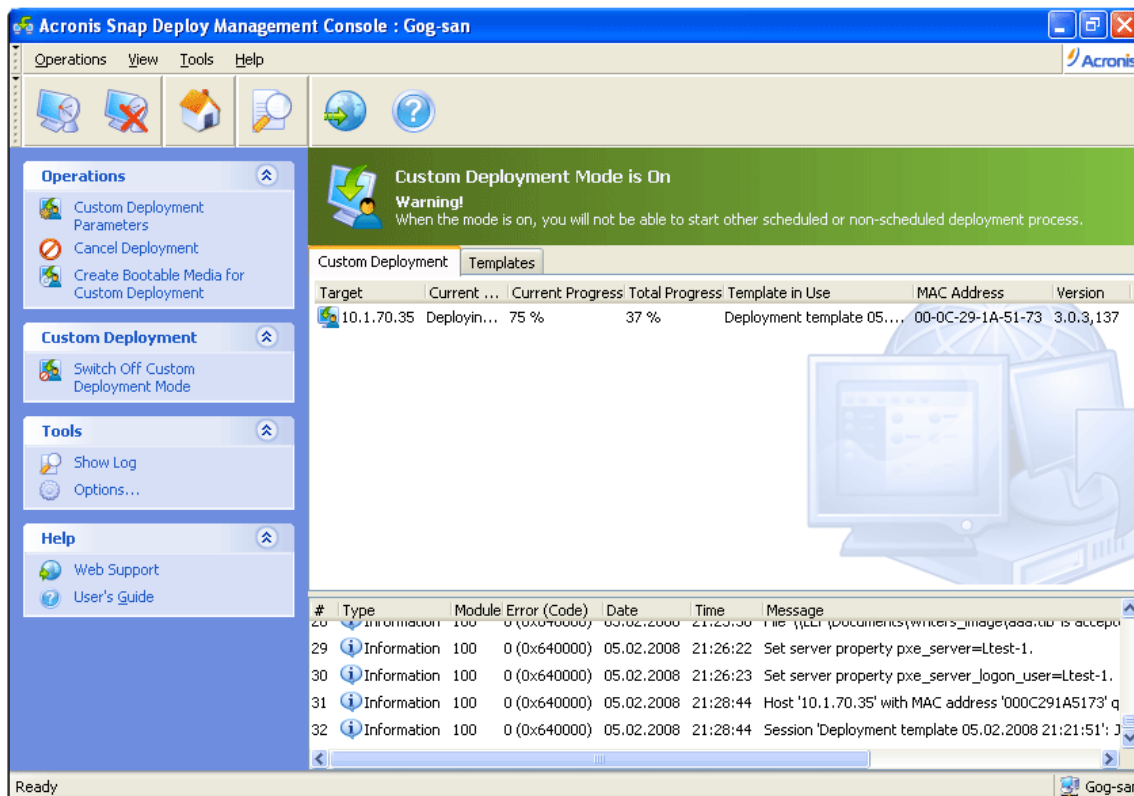


Figura 5: Consola de Acronis Snap Deploy 5.

3.1.6. System Center Configuration Manager (SCCM)

Microsoft System Center Configuration Manager 2012 (SCCM 2012) es un producto de Windows que permite a los administradores gestionar el despliegue y la seguridad de los dispositivos y aplicaciones en toda la empresa. SCCM es parte de la suite de gestión de sistemas 2012 [11].

Características: [12]

- Gestión de Windows 10. SCCM está diseñado para ir a la par con las actualizaciones de Windows 10 para soportar las nuevas características de Windows a medida que estén disponibles. Cuando SCCM se integra con Microsoft Intune, se puede elegir entre varias opciones de implementación y administración de Windows 10.
- Actualizaciones desde la consola. Con la creciente velocidad de las actualizaciones de Windows, iOS y Android, la solución de administración tiene que ser capaz de mantenerse al día también. Las actualizaciones y servicio de nodo de la consola de SCCM proporciona frecuentemente y de manera más fácil el aplicar las actualizaciones para las obtener nuevas características, actualizaciones acumulativas, extensiones para Microsoft Intune y correcciones individuales.

- Distribución de aplicaciones. SCCM utiliza un enfoque centrado en el usuario para la distribución de aplicaciones que permite a los administradores crear una aplicación que se puede entregar a todos los dispositivos de un usuario. Evalúa las capacidades del dispositivo y de la red y optimiza la entrega, ya sea a través de una instalación local, a través de App-V, o a través de un servidor, como los servicios de aplicaciones remotas de escritorio remoto o Citrix XenApp.
- Gestión de dispositivos. La integración con Microsoft Intune proporciona una única consola de administración para la gestión de políticas y activos, y de los informes de cumplimiento en ordenadores personales, así como los dispositivos móviles, incluyendo Windows, iOS y Android. Con las nuevas capacidades de gestión de dispositivos móviles (MDM Mobile Device Management), los dispositivos que ejecutan Windows 10 pueden ahora ser también administrados a través de MDM.
- Administración de escritorio virtual. SCCM reduce la complejidad y el costo de la implementación de entornos virtuales, proporcionando una única herramienta unificada para gestionar todos los escritorios de los clientes, clientes ligeros, dispositivos móviles y de escritorio virtuales.
- Endpoint Protection. SCCM se utiliza como infraestructura para System Center Endpoint Protection. Ofrece una solución única para la protección contra el *malware*, identificación y solución de vulnerabilidades.
- Gestión de la conformidad y ajustes. Puede crear una línea de base para el estado de configuración deseada y asegurar de que todos los dispositivos la cumplen través de un forzado automático o mediante alertas. SCCM también se integra con System Center Service Manager para crear automáticamente incidentes en la línea creada de referencia.
- Gestión de actualizaciones de software. SCCM simplifica la compleja tarea de distribuir y gestionar los cambios a los sistemas de TI en toda la empresa. Los administradores de TI pueden ofrecer actualizaciones de productos de Microsoft, aplicaciones de terceros, controladores de hardware y BIOS del sistema para una variedad de dispositivos, incluyendo ordenadores de sobremesa, portátiles, servidores y dispositivos móviles.
- Administración de energía. Permite sacar más provecho a las opciones de ahorro de energía del hardware, con un conjunto completo de herramientas centralizadas de administración de energía del cliente. SCCM trabaja con las capacidades integradas en el sistema operativo de Windows para ayudarle a optimizar la configuración de energía a nivel granular.
- Implementación del sistema operativo. SCCM distribuye sistemas operativos para ordenadores de sobremesa, servidores físicos, portátiles y dispositivos móviles a través de redes empresariales y elimina las ineficiencias y los errores asociados con la instalación de aplicaciones de forma manual. Con



Windows 10, SCCM también puede gestionar las actualizaciones *in situ*, que reducen significativamente el tiempo y la complejidad de la implementación de Windows.

- Monitorización y estado de salud de los clientes. SCCM muestra resultados de la evaluación de la salud y actividades de los clientes directamente en la consola, proporcionando alertas y funciones de corrección si las estadísticas de salud caen por debajo de los umbrales establecidos.
- Tratamiento inteligente de recursos. Los administradores tienen visibilidad continua sobre los recursos de hardware y software y de su utilización. El tratamiento inteligente de recursos convierte los datos inventariados en información útil, proporcionando informes que ayudan a los administradores con las decisiones de compra de software, planes de actualización, e informes de licencia.
- Inventario. SCCM puede hacer inventario de hardware y software en su organización para ayudar a darle una visión de los recursos que tiene. Con SCCM, puede activar el inventario de hardware a la medida con mayor facilidad y ampliar el esquema de inventario.
- Informes. Los informes en SCCM ayudan a recopilar, organizar y presentar la información acerca de: los usuarios, el inventario de hardware y software, actualizaciones de software, aplicaciones, el estado del sitio, y otras operaciones de SCCM.

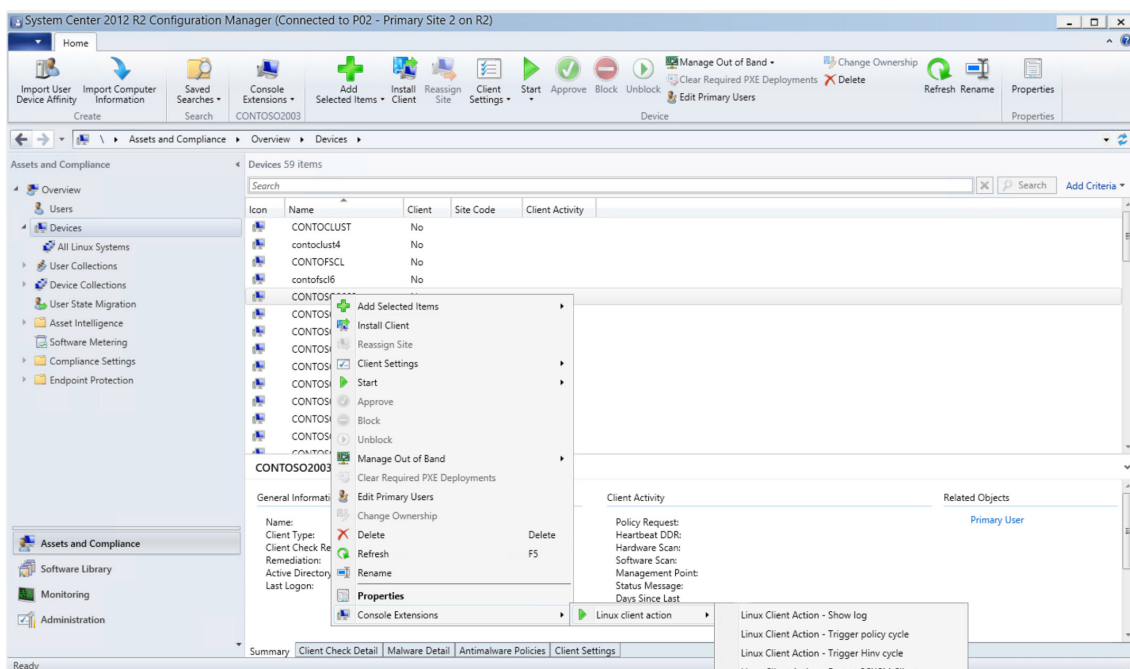


Figura 6: La consola de SCCM.

3.2. Comparativa de las herramientas

Seguidamente se comparan las principales características de las seis herramientas de despliegue de imágenes personalizadas del sistema operativo.

Tabla 1: Comparativa de las características de las seis herramientas.

	OpenGnSys	WDS	Ghost 3.1	MDT	Snap 5	SCCM
Se instala en SO Windows	No	Sí	Sí	Sí	Sí	Sí
Consola centralizada	Sí	Sí	Sí	Sí	Sí	Sí
Despliega LINUX	Sí	No	Sí	No	Sí	Sí
Despliega Windows	Sí	Sí	Sí	Sí	Sí	Sí
Despliega OSX	Sí	No	Sí	No	No	Sí
Soporte para IOS y Android	No	No	No	No	No	Sí
Programación de despliegue	Sí	No	Sí	Sí*	Sí	Sí
Controladores de dispositivos	No	Sí	Sí	Sí	Sí	Sí
Instalación de aplicaciones	No	Sí	Sí	Sí	Sí	Sí
Despliega sobre distinto hardware	No	Sí	Sí	Sí	Sí	Sí
Cliente-Servidor	Sí	No	Sí	No	Sí	Sí
Migración datos usuario	No	No	Sí	Sí	No	Sí
Permite Wake-on-Lan	Sí	No	Sí	No	Sí	Sí
Realiza inventario	Sí	No	Sí	No	No	Sí
Despliegue 100% desatendido	Sí	Sí	Sí	Sí*	Sí	Sí
Configuración individual de un ordenador	Sí	Sí	Sí	Sí	Sí	Sí
Importe de 500 licencias	Gratis	Gratis	8 046 €	Gratis	8 712 €	53 240 €

* MDT permite hacer despliegues desatendidos y programados, añadiendo configuración adicional y utilizando SQL Server.

Atendiendo estrictamente a sus características, las mejores herramientas son:

- Microsoft System Center Configuration Manager (SCCM).
- Symantec Ghost Solution Suite 3.1.
- Acronis Snap Deploy 5.

Cualquiera de estas tres herramientas cumple a la perfección las necesidades de despliegue de imágenes personalizadas en un entorno corporativo como el de la Universitat Politècnica de València. Las tres herramientas presentan el inconveniente del alto precio de las licencias de uso. Dada la infraestructura administrativa de la UPV, a una escuela le es complicado el negociar el importe de dichas licencias.

Después de un exhaustivo análisis de todas las herramientas, y considerando las cuestiones económicas, se decide utilizar **Microsoft Deployment Toolkit (MDT)**. MDT es la mejor herramienta gratuita que cumple con las necesidades de la escuela para el despliegue de imágenes personalizadas del sistema operativo. Su integración con los

sistemas informáticos actuales de la escuela, el que su licencia sea gratuita y su flexibilidad y capacidad de expansión por medio de herramientas externas, hacen que sea la mejor opción.

4. Microsoft Deployment Toolkit (MDT)

Ahora que tenemos seleccionada la herramienta, se va a analizar y describir con más detalle su funcionamiento para conseguir realizar despliegues automatizados.

4.1. Descripción de la herramienta MDT

Microsoft Deployment Toolkit (MDT) es un producto hecho por Microsoft para ayudar al despliegue del sistema operativo Windows. Se puede utilizar en conjunción con otros productos de Microsoft como el WAIK/ADK, USMT, PXE/WDS, WSIM, Windows OOBE setup, y System Center Configuration Manager, y proporciona una interfaz y un entorno unificado para todas herramientas anteriores. Tiene una comunidad muy amplia de profesionales de TI que lo utilizan. Es una herramienta gratuita.

MDT ofrece un entorno ligero basado en secuencias de tareas que permiten desplegar imágenes de Windows, o realizar otras tareas simples [13].

MDT proporciona una colección unificada de herramientas, procesos y guías para la automatización de las implementaciones de ordenadores de escritorio y servidores. Además de reducir el tiempo de implementación y estandarizar las imágenes de ordenadores de escritorio y servidor, MDT ofrece opciones mejoradas de administración de la seguridad y de configuración [14].

4.1.1. Conceptos previos

MDT tiene varias maneras de desplegar las imágenes basándose en el grado de interacción con el usuario:

LiteTouch (LTI) es un término que describe un proceso de despliegue de MDT en el que el usuario tiene que interactuar mínimamente al principio del despliegue, y el resto del proceso es automático.

ZeroTouch (ZTI) hace referencia a un proceso de despliegue de SCCM con las extensiones de MDT. Desde la perspectiva del usuario el proceso es totalmente automático, no requiere ninguna interacción.

User Driven Installation (UDI) es un asistente que permite proporcionar información de configuración inmediatamente antes de realizar la implementación. Este comportamiento le permite crear secuencias de tareas de despliegue de sistema operativo (OSD) genéricos, y luego proporciona información específica del equipo en el momento del despliegue, que permite una mayor flexibilidad en el proceso de implementación.

La manera de llamar a los tipos de despliegue según el nivel de interacción con el usuario, es una cuestión de nomenclatura y puede llegar a ser confuso, pues un despliegue LTI se puede configurar para que sea ZTI, y al revés, un despliegue ZTI se puede hacer LTI.

4.1.2. Tipos de despliegue. La elección de LTI, ZTI, o UDI

Las implementaciones LTI, ZTI, y UDI utilizan el mismo conjunto común de secuencias de comandos y archivos de configuración (como CustomSettings.ini) para el despliegue de los equipos de destino [15].

Tabla 2: Comparativa de los tres tipos de despliegues: LTI, ZTI y UDI.

LTI	ZTI	UDI
Permite elegir el nivel de automatización	Solo soporta despliegues totalmente automáticos	Permite elegir el nivel de automatización
Requiere de una infraestructura mínima	Requiere SCCM	
Es compatible con las implementaciones vía red utilizando una carpeta compartida o localmente utilizando almacenamiento extraíble, como un CD, un DVD o UDF	Es compatible con las implementaciones vía red utilizando los puntos de distribución de SCCM o localmente utilizando almacenamiento extraíble, como un CD, un DVD o UDF	
El proceso de implementación se puede iniciar manualmente o automáticamente mediante los Servicios de implementación de Windows	El proceso de instalación puede ser iniciado por SCCM o los Servicios de implementación de Windows	
El proceso de implementación se configura con el Deployment Workbench	El proceso de implementación se configura utilizando la consola de SCCM	El proceso de implementación se configura utilizando la consola de SCCM y el asistente de diseño UDI
Requiere menos tiempo de administración de TI para la configuración inicial	Requiere más tiempo para la configuración inicial	
Puede requerir la interacción del usuario o de un técnico en el despliegue	No necesita ninguna interacción del usuario o de un técnico en el despliegue	Puede requerir la interacción del usuario o de un técnico en el despliegue
Aumenta el riesgo de introducir errores de configuración	Reduce el riesgo de introducir errores de configuración	Aumenta el riesgo de introducir errores de configuración
Requiere que los usuarios o técnicos de implementación tengan credenciales con permisos elevados	No requiere que los usuarios o técnicos de implementación tengan credenciales con permisos elevados	Requiere que los usuarios o técnicos de implementación tengan credenciales con permisos elevados
Requiere que los usuarios o técnicos de implementación conozcan algunos valores de configuración antes de iniciar el proceso de implementación de MDT	Usuarios y los técnicos de implementación no necesitan saber los valores de configuración antes de iniciar el proceso de implementación de MDT	Requiere que los usuarios o técnicos de implementación conozcan algunos valores de configuración antes de iniciar el proceso de implementación de MDT

Se puede utilizar con conexiones lentas o en casos en los que no hay conectividad de red	Requiere una conexión persistente de alta velocidad
Requiere poca o ninguna infraestructura para realizar el despliegue	Requiere de una infraestructura mínima para implementar imágenes del sistema operativo
Admite la implementación vía red o en local utilizando algún medio: USB, DVD	Admite la implementación vía red o en local utilizando algún medio: USB, DVD
No requiere gestión del equipo de destino mediante SCCM	Requiere gestión del equipo de destino mediante SCCM
Soporta políticas de seguridad en el que se prohíbe la instalación automática de software	Sólo soporta seguridad en la que se permite la instalación automática de software.
Admite la implementación en equipos de destino aislados por los cortafuegos	Requiere comunicación con la llamada a procedimiento remoto (RPC) con los equipos de destino

En algún momento del proceso de MDT, se debe proporcionar toda la información necesaria para instalar Windows y las aplicaciones en los equipos de destino. La pregunta es, ¿cuándo proporcionar esta información? Cuanta más información se proporcione de antemano, menor interacción se requerirá durante el despliegue.

Tabla 3: Ventajas y desventajas de los despliegues totalmente automatizados y parcialmente automatizados.

Método	Ventajas	Desventajas
Totalmente automatizado	<ul style="list-style-type: none"> No se requiere ninguna interacción con el usuario o técnico de despliegue. El riesgo de introducir errores de configuración se reduce. Usuarios o técnicos de implementación no necesita saber nada de la información de configuración antes de iniciar el proceso de implementación de MDT. 	<ul style="list-style-type: none"> Se necesita más tiempo para proporcionar información de configuración necesaria para el despliegue totalmente automatizado. Las credenciales necesarias para acceder a los recursos de red y que requieren de permisos elevados, se almacenan en archivos de configuración que deben ser protegidos.
Parcialmente automatizado	<ul style="list-style-type: none"> Se requiere menos tiempo para prepararse el despliegue, ya que la información de configuración se puede proporcionar de forma interactiva. 	<ul style="list-style-type: none"> Se requiere interacción con el usuario o técnico de despliegue. El riesgo de introducir errores de configuración se incrementa. Usuarios o técnicos de implementación debe tener credenciales que requieren permisos elevados. Usuarios o técnicos de implementación deben conocer alguna información de configuración antes de iniciar el proceso de implementación de MDT.



4.1.3. ¿Qué son las imágenes? ¿Qué es SysPrep?

Desde el sistema operativo Windows Vista, Microsoft se ha preocupado de preparar un método que permita desplegar una imagen de sus sistemas operativos en un disco duro. Esto significa que una vez tenemos un ordenador preparado en el estado que queremos, es decir, con las aplicaciones que necesitamos instaladas, con todos los ficheros que necesitamos y con la configuración que nos interesa, el disco es copiado a un fichero de imagen que puede ser desplegado a otros ordenadores [13].

La manera de preparar un ordenador para ser clonado es ejecutando la herramienta SysPrep.exe justo antes de capturar la imagen. La imagen que se obtiene se almacena en uno o varios ficheros con la extensión WIM (Windows IMaging). Son ficheros comprimidos y tiene el mismo formato que podemos encontrar en los DVD de distribución de Windows.

MDT se puede usar en dos escenarios diferentes: capturar imágenes y desplegar imágenes.

- Capturar imagen: es el proceso de crear un nuevo fichero *.wim que contiene aplicaciones, configuraciones y a veces algunos controladores. Idealmente, MDT puede instalar automáticamente el sistema operativo, las aplicaciones, configuraciones específica en el registro del sistema, actualizarlo desde Windows Update, ejecutar la herramienta SysPrep.exe y guardar el contenido del disco duro en un fichero *.wim.
- Desplegar imagen: es el proceso de instalar en un ordenador un fichero *.wim, que puede ser una versión de distribución de Windows o una imagen personalizada. MDT permite además instalar aplicaciones adicionales, aplicar configuraciones, instalar controladores y migrar los ficheros y configuración de los usuarios.

Como tenemos dos opciones para instalar las aplicaciones que necesitamos, surge la siguiente pregunta: ¿Instalar las aplicaciones directamente en la imagen?, o ¿instalarlas en el momento de hacer el despliegue? Hay que considerar algunos factores para determinar la respuesta correcta.

- Tamaño de las aplicaciones a instalar después de haber realizado el despliegue.
- Si la aplicación requiere interactuar con el usuario para su instalación, como, por ejemplo, solicitar algún tipo de activación o licencia.
- Si la aplicación tiene que ser instalada en todos los ordenadores que reciban la imagen.

4.1.4. Tamaño de las imágenes: pesadas, ligeras e híbridas

Como parte del proceso de planificación, hay que determinar el tipo de las imágenes que vamos a crear. Las imágenes pueden ser:

- **Pesadas.** Imágenes pesadas son imágenes monolíticas que contienen aplicaciones, paquetes de idiomas y otros archivos. Parte del proceso de desarrollo de la imagen es la instalación de las aplicaciones principales y paquetes de idioma en el equipo de referencia antes de capturar la imagen de disco.
- **Ligeras.** Las imágenes ligeras contienen pocas o ninguna aplicación o paquetes de idioma, ya que estos componentes se instalan por separado de la imagen de disco, que por lo general toma más tiempo de transferencia de red en el equipo.
- **Híbridas.** En la imagen híbrida se mezclan las estrategias de imagen ligera y pesada mediante la instalación de aplicaciones y paquetes de idioma desde una carpeta de red compartida. Las imágenes híbridas tienen la mayor parte de las ventajas de las imágenes ligeras, pero no son tan complejas de desarrollar y no requieren una infraestructura de distribución de software. Pueden requerir largos tiempos de instalación, lo que puede elevar los costes del despliegue inicial.

Tabla 4: Ventajas y desventajas del tamaño de las imágenes.

	Ventajas	Desventajas
Pesada	<ul style="list-style-type: none"> • Será más fácil de implementar, ya que todas las aplicaciones y paquetes de idioma están en la imagen. • Reducción de la complejidad inicial, porque no hace falta realizar <i>scripting</i> avanzado. • Las aplicaciones y los paquetes de idioma están disponibles inmediatamente después del despliegue. • No requiere distribución de software, tales como la característica de administración de aplicaciones de SCCM. 	<ul style="list-style-type: none"> • Consume más espacio de almacenamiento para cada imagen. • Utiliza más tiempo para descargar a través de conexiones de red que las imágenes finas o híbridos. • Requiere un esfuerzo extra el mantenimiento de la imagen. Debido a las actualizaciones de los sistemas operativos, controladores de dispositivos, aplicaciones o paquetes de idioma, puede requerir el tener que crear una nueva imagen.
Ligera	<ul style="list-style-type: none"> • Utiliza menos espacio de almacenamiento para cada imagen. • Invierte menos tiempo que las imágenes pesadas en la descarga a través de conexiones de red. • Reduce el esfuerzo de mantenimiento de la imagen, ya que la imagen contiene menos componentes. 	<ul style="list-style-type: none"> • Puede ser más compleja inicialmente, ya que se requieren pasos adicionales durante la creación de la imagen. • Mayor complejidad, puede hacer falta <i>scripting</i> avanzado. • Las aplicaciones y paquetes de idiomas no están disponibles inmediatamente después de la implementación.

Híbrida	<ul style="list-style-type: none"> • Menos espacio de almacenamiento que las imágenes pesadas para cada imagen. • Menos tiempo para que una imagen pesada para descargar a través de conexiones de red. • Reducción del esfuerzo de mantenimiento de la imagen, ya que la imagen contiene menos componentes. • No requiere de distribución de software adicional. 	<ul style="list-style-type: none"> • Puede ser más compleja de crear que una imagen pesada (pero no de una imagen ligera), ya que se requieren pasos adicionales durante la creación de la imagen. • Posibilidad de una mayor complejidad, ya que puede ser necesario utilizar <i>scripting</i> avanzado (aunque no tan avanzado como en imágenes ligeras). • Las aplicaciones y paquetes de idiomas no están disponibles inmediatamente después de la implementación.
---------	---	---

Los costos asociados con la creación, mantenimiento y despliegue de imágenes de disco incluyen:

- Los costos de desarrollo. Los costos de desarrollo incluyen la creación de una imagen utilizando buenas practicas, para reducir los futuros costes de soporte, mejorar la seguridad y la fiabilidad. Los niveles más altos de automatización reducen los costes de desarrollo.
- Costes de las pruebas. Estos costes incluyen el tiempo y la mano de obra involucrada en la prueba de la imagen estándar y las aplicaciones que pueden residir dentro de ella, además de aplicaciones instaladas después del despliegue. También incluyen el tiempo de desarrollo necesario para estabilizar las imágenes de disco.
- Costos de almacenamiento. Incluyen el almacenamiento de los puntos de distribución, imágenes de disco, los datos de migración, y las imágenes de copia de seguridad. Pueden ser significativos en función del número de imágenes de disco, el número de equipos en cada ejecución de despliegue, y así sucesivamente.
- Los costos de red. Incluye el movimiento de imágenes a puntos de distribución y hacia los ordenadores cliente. Las tecnologías de imágenes de disco que ofrece Microsoft no admiten la multidifusión, por lo que los costes de red aumentan linealmente con el número de puntos de distribución que hay que replicar y el número de ordenadores en el que se implementan.

A medida que el tamaño de los archivos de imagen aumenta, los costos aumentan. Las imágenes grandes tienen más costes asociados dado que requieren un mayor número de actualizaciones, pruebas, distribución, y consumo de ancho de banda de red. Incluso si se actualiza sólo una pequeña parte de la imagen, la imagen completa debe ser redistribuida de nuevo.

En la ETSIAMN, para el despliegue de imágenes personalizadas, se utilizaron imágenes híbridas, es decir, una combinación de imagen pesada con instalación de aplicaciones posterior al despliegue. Se creó una imagen pesada en el ordenador de referencia, donde se instaló lo siguiente:

1. El sistema operativo.
2. Las aplicaciones.
3. Se personalizó el escritorio con iconos de acceso a aplicaciones para todos los usuarios.
4. Se actualizó el sistema operativo y las aplicaciones instaladas.

Cuando la imagen se despliega en un ordenador cliente, se instaló el software de antivirus y las aplicaciones específicas que requiera cada ordenador como, por ejemplo, un software de control de un escáner.

4.2. Requisitos de software para MDT

Para que MDT funcione correctamente y podamos realizar despliegues de imágenes ZTI, además de instalar el propio MDT, hay que instalar el siguiente software:

1. Net Framework 3.5
2. Windows 10 ADK versión 1511
3. SQL Server 2014 Express Advanced

Vamos a ver cuáles son los sistemas operativos soportados y el software que necesita para funcionar correctamente.

4.2.1. Sistemas operativos soportados

Los sistemas operativos soportados y donde se puede instalar MDT son los siguientes: Windows 7, Windows 8, Windows 8.1, Windows 10, Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2.

4.2.2. ADK de Windows 10

Windows ADK de Windows 10, contiene las herramientas que necesita MDT para poder realizar el despliegue de imágenes. Solo el ADK no es suficiente para poder realizar el despliegue.

4.2.3. SQL 2014 Express Advanced

Para conseguir realizar despliegue de imágenes utilizando ZTI en MDT, es necesario instalar una distribución de SQL Server, que nos permitirá crear una base de datos donde MDT almacenara los datos de los ordenadores donde se desplegaran las imágenes, como, por ejemplo:

- El nombre del ordenador.
- El dominio o el grupo de trabajo donde se integrará el ordenador.
- La resolución de la pantalla.
- El UUID del ordenador.



La versión de SQL Server a instalar puede ser la SQL Express, que aunque tiene inferiores características respecto de la versión SQL Server, es más que suficiente para soportar la base de datos que necesitamos para almacenar los datos de los ordenadores que utilizaremos con MDT. En nuestro caso la base de datos contiene actualmente cuatrocientos veinticinco registros y ocupa únicamente 6,38 MB de espacio de disco. SQL Express es una edición gratuita y no requiere adquirir ninguna licencia para su utilización.

Tabla 5: Resumen comparativo de las ediciones de Microsoft SQL Server.

Características	Enterprise	Business Intelligence	Standard	Express
Número máximo de núcleos	SO máximo	16 núcleos, SO de base de datos máx. AS&RS2	16 núcleos	4 núcleos
Memoria máxima usada por instancia	SO máximo	128 GB	128 GB	1 GB
Tamaño máximo de la base de datos	524 PB	524 PB	524 PB	10 GB

Podemos consultar todas las características de las diferentes versiones de Microsoft SQL Server en la siguiente página web:

<https://www.microsoft.com/es-es/server-cloud/products/sql-server-editions/overview.aspx>

4.2.4. Software que hay que instalar

Net Framework 3.5. Se puede instalar como una característica del sistema operativo, o bien descargarlo del siguiente enlace:

<https://www.microsoft.com/es-ES/download/details.aspx?id=21>

Windows 10 ADK, la versión actual es la 1511, la página para su descarga:

<https://msdn.microsoft.com/es-es/windows/hardware/dn913721.aspx>

Microsoft Deployment Toolkit 2013 Update 2

<https://www.microsoft.com/en-us/download/details.aspx?id=50407>

SQL Server 2014 Express Advanced

<https://www.microsoft.com/es-es/download/details.aspx?id=42299>

<https://technet.microsoft.com/en-us/itpro/windows/deploy/windows-deployment-scenarios-and-tools>

4.2.4.1. Orden de instalación del software

Windows 10 ADK 1511 incorpora diversas herramientas:

- Kit de compatibilidad de aplicaciones (ACT).
- Herramientas de implementación.
- Entorno de preinstalación de Windows (Windows PE).
- Diseñador de imágenes y configuraciones (ICD).
- Herramienta de migración de estado de usuario (USMT)
- Volume Activation Management Tool (VAMT).
- Kit de herramientas de rendimiento de Windows.
- Servicios de evaluación de Windows.
- Microsoft SQL Server 2012 Express

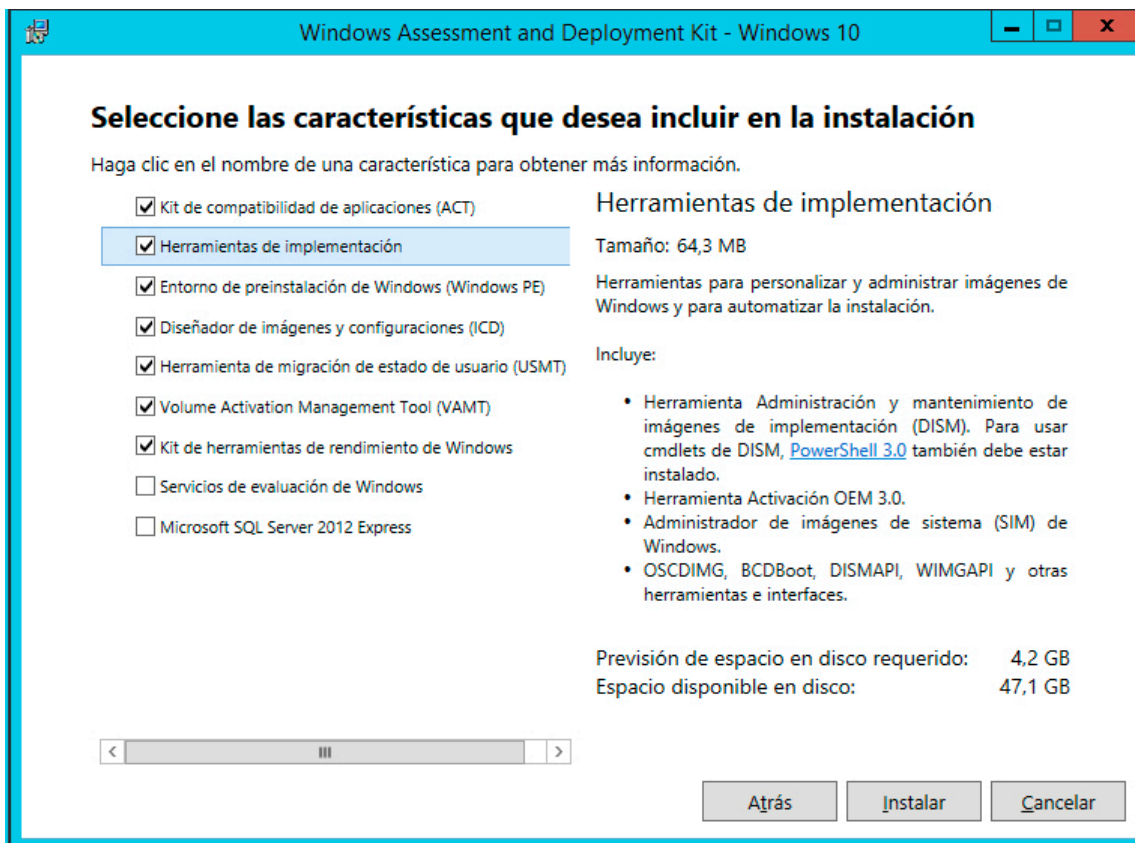


Figura 7: Pantalla de instalación de Windows ADK 1511.

Las características que necesitamos instalar necesariamente para que MDT funcione son dos:

- Herramientas de implementación. Son herramientas para personalizar y administrar imágenes de Windows y para automatizar la instalación.
- Entorno de preinstalación de Windows (Windows PE). Sistema operativo mínimo diseñado para preparar la instalación y mantenimiento de Windows.

Desmarcamos la instalación de SQL Server 2012 Express porque lo que haremos posteriormente es instalar la versión SQL Server 2014 Express Advanced. Cuando lo hagamos también se desmarcará la opción de Servicios de evaluación de Windows. El resto de herramientas de Windows ADK podemos optar por instalarlas o no en este momento. Si no las instalamos ahora, posteriormente las podremos instalar lanzando de nuevo la instalación del ADK.

Para la instalación de Microsoft Deployment Toolkit 2013 Update 2(MDT) simplemente hay que seguir el asistente.

Finalmente instalamos SQL 2014 Express Advanced. El motivo de elegir la versión *Advanced*, es porque incorpora SQL Server Management Studio (SSMS). Es un entorno integrado para obtener acceso, configurar, administrar y desarrollar todos los componentes de SQL Server. SSMS combina un amplio grupo de herramientas gráficas con una serie de editores de *script* avanzados que permiten a desarrolladores y administradores de todos los niveles obtener acceso a SQL Server [16].

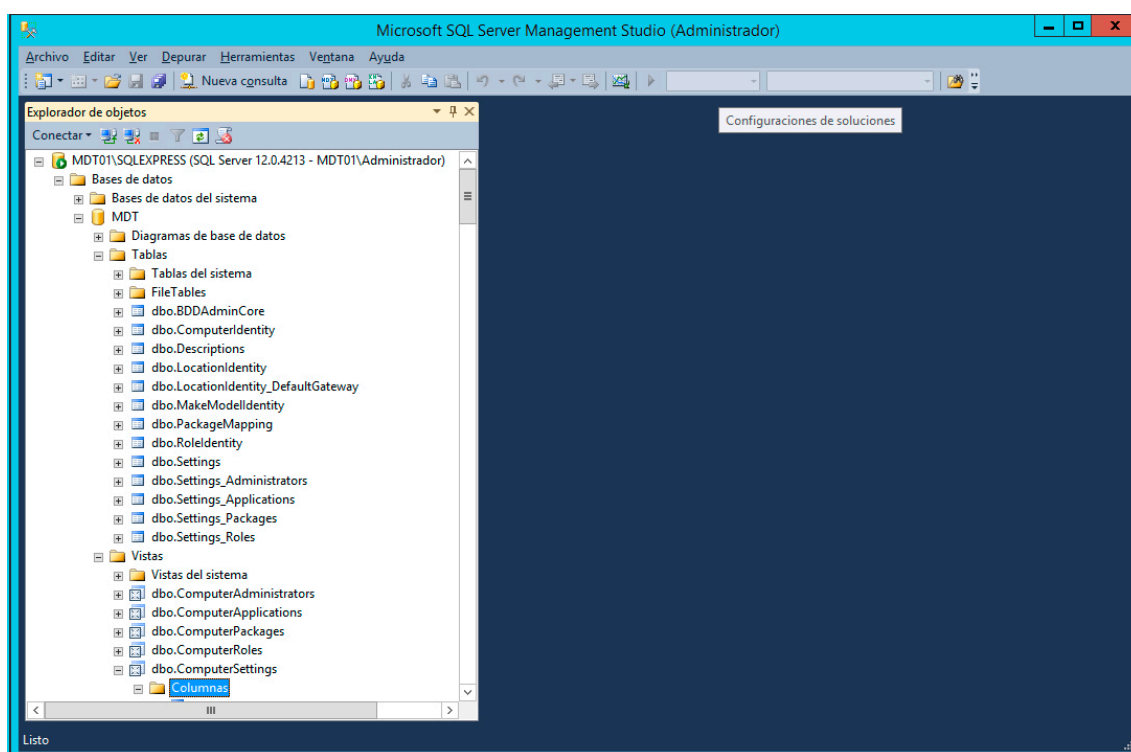


Figura 8: Consola de Microsoft SQL Server Management Studio.

Una vez instalado SQL Server 2014 Express, se debe de comprobar que el servicio SQL Server Browser esta iniciado, de lo contrario no será accesible desde otros ordenadores. Se comprueba desde la consola de administración de equipos, o desde la consola de SQL Server Configuration Manager.

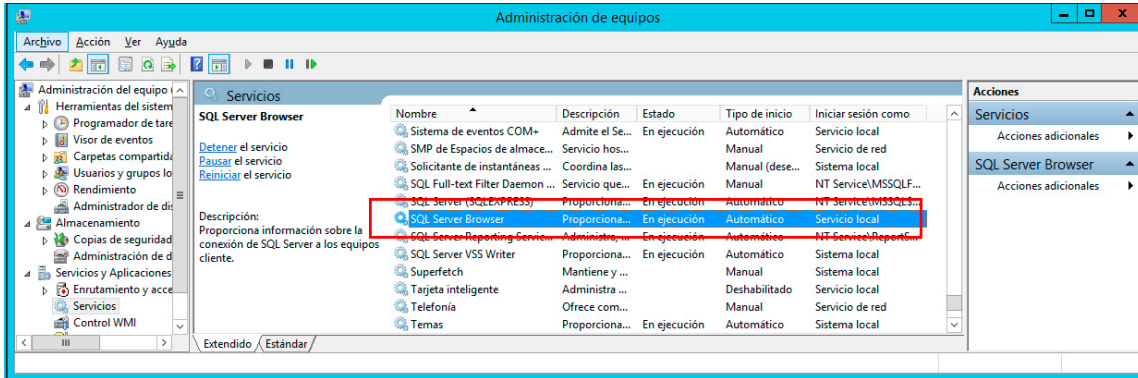


Figura 9: El servicio SQL Server Browser en la consola de administración de equipos.

4.3. Componentes de MDT

El entorno de MDT recibe el nombre de Deployment Workbench (DW), y consta de varios componentes que vamos a detallar a continuación [17].

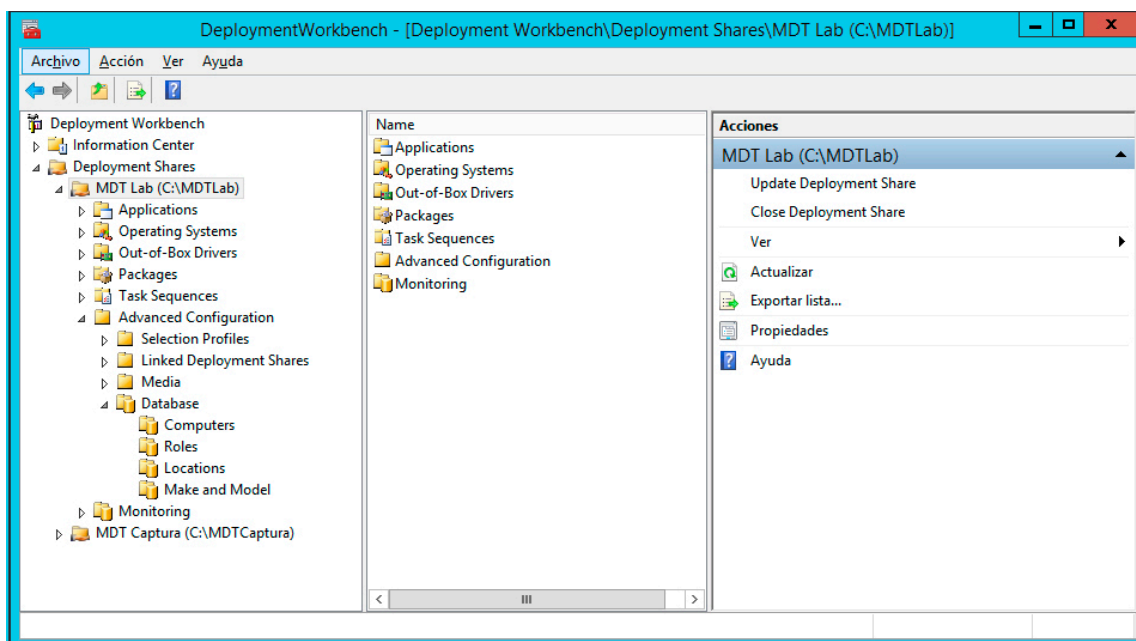


Figura 10: Entorno de MDT, el Deployment Workbench.

4.3.1. Deployment Share (DS) o punto de despliegue

Es esencialmente una carpeta compartida en un servidor y que contiene todos los archivos de configuración y ficheros de *scripts* necesarios para la implementación. También contiene los archivos de configuración (llamadas reglas) que se aplican cuando una máquina recibe un despliegue. Estos archivos de configuración pueden extenderse a otras fuentes, como una base de datos, un *script* externo, o un servidor web para obtener configuraciones adicionales para el despliegue. Para las implementaciones LTI,

es común tener dos procesos de implementación: uno para la creación de las imágenes de referencia y uno para el despliegue.

4.3.2. Rules o reglas

Realmente son los ficheros CustomSettings.ini y bootstrap.ini del DS y conforman el núcleo de MDT. Las reglas controlan el asistente para implementar Windows en la parte del cliente y puede proporcionar, por ejemplo, los siguientes ajustes al ordenador que recibe el despliegue:

- Nombre del ordenador.
- Dominio para unirse, y la unidad organizativa (OU) dentro de Active Directory (AD).
- Si se debe habilitar BitLocker.
- Ajustes regionales.

La configuración de las reglas dentro de cada DS, se aplican por defecto a todos los ordenadores que realicen el despliegue desde ese mismo DS, pero serán ignoradas si posteriormente se encuentran reglas específicas para el ordenador en la base de datos. Por ejemplo, si en las reglas del DS está configurado por defecto que la máquina se tiene que agregar al dominio A, pero un ordenador concreto tiene que agregarse al dominio B, y está configurado de esa manera en las propiedades específicas del ordenador dentro de la base de datos de MDT, prevalece la configuración específica del ordenador, y se agregará al dominio B, y no al dominio A.

4.3.3. Boot images o imágenes de arranque

Son imágenes basadas en el entorno de preinstalación de Windows (Windows PE) que se utilizan para iniciar el despliegue. Se pueden iniciar desde: un CD o un DVD, un archivo ISO, un dispositivo USB o en la red mediante un servidor de ejecución previo al inicio (PXE). Las imágenes de arranque se conectan al recurso compartido de implementación en el servidor e inician el despliegue.

4.3.4. Operating Systems o sistemas operativos

Utilizando el Deployment Workbench (DW) se importan los sistemas operativos que queremos implementar. Podemos importar el sistema operativo completo (como el DVD de distribución de cualquier sistema operativo Windows, que sea de la versión 7 o superior, o bien una imagen ISO del DVD), o una imagen personalizada que hayamos creado. Los sistemas operativos completos se utilizan principalmente para crear imágenes de referencia; sin embargo, también se pueden utilizar para las implementaciones normales.

4.3.5. Applications o aplicaciones

En el Deployment Workbench podemos también añadir las aplicaciones que deseamos implementar después del despliegue. MDT es compatible con prácticamente todos los tipos de archivo ejecutables. El archivo puede ser: un archivo .exe estándar con modificadores de línea de comandos para una instalación desatendida, un paquete de Microsoft Windows Installer (MSI), un archivo por lotes o un fichero *VBScript*. De hecho, puede ser casi cualquier fichero que se puede ejecutar desatendidamente.

4.3.6. Out-of-Box Drivers o repositorio de controladores de dispositivos

Deployment Workbench puede importar en su repositorio los controladores de dispositivo que necesita cada tipo de hardware, de modo que los controladores están en el servidor, no en la imagen.

4.3.7. Packages o paquetes

En el Deployment Workbench se puede añadir cualquier paquete de Microsoft que se desea utilizar. Los paquetes más comúnmente añadidos son paquetes de idioma. También se pueden añadir actualizaciones de seguridad y otras actualizaciones de esta manera. Sin embargo, generalmente se recomienda utilizar Windows Server Update Services (WSUS) para las actualizaciones del sistema operativo. Las excepciones más raras son las revisiones de críticas que no están disponibles a través de WSUS, paquetes para las imágenes de arranque, o cualquier otro paquete que debe ser desplegado antes de que comience el proceso de actualización de WSUS.

4.3.8. Task Sequences (TS) o secuencias de tareas

Son el corazón y el alma de la solución de implementación. Para crear una secuencia de tareas nueva, tenemos que crearla a partir de una plantilla. Las plantillas se encuentran en la carpeta Templates en el directorio de instalación de MDT, y determinan qué acciones por defecto están presentes en la secuencia.

Se puede pensar en una secuencia de tareas como una lista de acciones que deben ser ejecutadas en un orden determinado. Cada acción también puede tener condiciones. Algunos ejemplos de acciones son las siguientes:

- Recopilar (*Gather*). Lee los valores de configuración del servidor de implementación.
- Formato y partición. Crea las particiones y las formatea.
- Inyectar controladores de dispositivos. Busca los controladores de dispositivos que necesita el ordenador y los descarga desde el repositorio de controladores.
- Aplicar el sistema operativo. Utiliza DISM para aplicar la imagen.



- Actualización de Windows. Se conecta a un servidor WSUS y actualiza la máquina.

4.3.8.1. Plantillas de secuencias Task Sequence Templates

MDT tiene once plantillas de secuencias por defecto:

- **Sysprep and Capture task sequence.** Ejecuta la herramienta Sysprep y realiza la captura de una imagen del ordenador de referencia.
- **Standard Client task sequence.** Se usa tanto para crear imágenes de referencia como para desplegar imágenes.
- **Standard Client Replace task sequence.** Se utiliza para reemplazar un ordenador haciendo una copia de seguridad de los datos de un cliente (USMT).
- **Standard Client Upgrade task sequence.** Permite actualizar un ordenador a Windows 10.
- **Custom task sequence.** Secuencia personalizada que solo contiene una orden, como por ejemplo instalar una aplicación.
- **Litetouch OEM task sequence.** Precarga el Sistema operativo en el disco duro del ordenador. Lo utilizan los fabricantes de ordenadores.
- **Standard Server task sequence.** Secuencia por defecto para desplegar una imagen de un servidor. Esta secuencia no guarda ningún dato de usuario, ya que esta acción no está soportada en un servidor.
- **Standard Server Upgrade task sequence.** Únicamente para actualizar a Windows Server 2016.
- **Post OS Installation task sequence.** Esta secuencia permite realizar acciones después de que la imagen se haya desplegado.
- **Deploy to VHD Client task sequence.** Es parecida a la Standard Client task, pero además crea un disco duro virtual (VHD) en el ordenador cliente y despliega la imagen en el VHD.
- **Deploy to VHD Server task sequence.** *Ídem* que la secuencia Deploy to VHD Client task sequence, pero para servidores.

4.3.9. Selection Profiles o perfiles de selección

Están disponibles en la carpeta de configuración avanzada, proporcionan una manera de filtrar el contenido del Deployment Workbench. Los perfiles de selección se utilizan con varios fines en las implementaciones que utilizan LTI. Por ejemplo, se pueden utilizar para controlar:

- Que controladores de dispositivos y paquetes se inyectan en el Lite Touch, y en imágenes de arranque genéricas.
- Que controladores de dispositivos se inyectan durante la secuencia de tareas.
- Que se incluye en cualquier medio que se cree.

- Que se replica en otros puntos de implementación (DS).
- Filtrar que secuencias de tareas y aplicaciones se muestran en el asistente para la implementación.

4.3.10. Logging o ficheros de registro

MDT utiliza muchos archivos de registro durante las implementaciones del sistema operativo. Por defecto los registros se crean en la parte del cliente, pero configurando los parámetros de implementación se pueden tener también almacenados en el servidor.

4.3.11. Monitoring o monitorización

Activando la opción de monitorización del DS podemos monitorizar el desarrollo de despliegues en el Deployment Workbench.

4.4. Ejemplo práctico HELLO WORLD

Para realizar despliegues con MDT, es necesario tener:

- un dominio en un entorno corporativo, por ejemplo: alumno
- un servidor de nombre MDT01, ejecutando Windows Server 2012 R2 e integrado en el dominio alumno
- un usuario del dominio alumno, con privilegios de administrador en todos los ordenadores, y que nuestro caso será el usuario AdminMDT
- el siguiente software instalado en el servidor MDT01:
 1. Windows 10 ADK 1511
 2. MDT 2013 Update 2
 3. SQL 2014 Express

Estos son los pasos que se siguieron:

1. Se creó un DS (si no teníamos ninguno creado) en MDT.
2. Se creó un usuario que no es administrador del ordenador, pero tiene que tendrá permisos de lectura y de ejecución en el recurso compartido del punto de despliegue.
3. Se importó un sistema operativo.
4. Se creó una secuencia de tareas para realizar el despliegue. MDT incorpora plantillas de secuencias de tareas listas para funcionar. Sin más que crear una nueva secuencia y sin realizar ninguna modificación, podemos realizar un despliegue LTI.
5. Se realizó un despliegue LTI, ejecutando la secuencia de tareas del punto 4, para comprobar que funcionaba correctamente.

Notas:

- El paso número 1 hay hacerlo cada vez que queramos crear un nuevo DS. En nuestro caso, la idea es tener al final, tres DS:
 1. un DS para capturar imágenes de referencia, MDTCaptura.
 2. un DS para realizar pruebas, MDTLab.
 3. un DS para desplegar las imágenes en producción, MDTProduccion.
- El paso número 2 solo hay que hacerlo una vez. Una vez creado el usuario no hay que crearlo de nuevo cuando se crea un nuevo DS.

4.4.1. Paso número 1: crear un Deployment Share

Para la creación de un nuevo punto de despliegue (Deployment Share), pulsamos con el botón derecho del ratón sobre la opción Deployment Shares y seleccionamos New Deployment Share. Se muestra un asistente para ir seleccionando las diversas opciones de creación del punto de despliegue. Tenemos que seguir las indicaciones del asistente en cada pantalla.

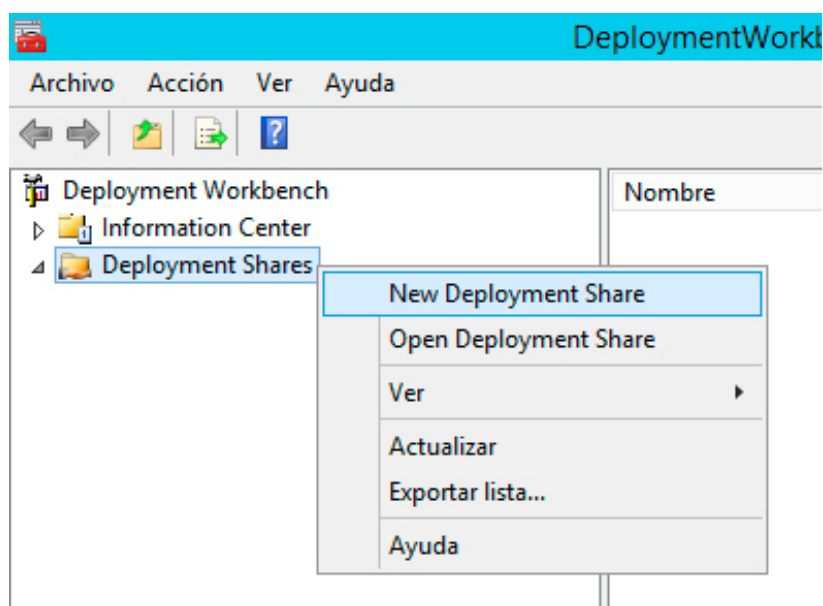


Figura 11: Como crear un nuevo Deployment Share.

Hay que especificar la ubicación del nuevo punto de despliegue. Microsoft recomienda crear los puntos de despliegue en cualquier unidad distinta a la que utiliza el sistema operativo, pero como en nuestro caso se trata de un punto de despliegue para realizar pruebas, lo creamos en la misma unidad donde está el sistema operativo. Escribimos en la ruta de creación C:\MDTLab.

Path	Specify the location of the network share for this deployment type.
Share	
Descriptive Name	
Options	
Summary	
Progress	
Confirmation	

Deployment share path:	<input type="text" value="C:\MDTLab"/>	<input type="button" value="Browse..."/>
------------------------	--	--

Figura 12: La carpeta donde se creará el Deployment Share.

Hubo que asignar un nombre al recurso compartido del punto de despliegue. Podíamos haberle dado cualquier nombre, pero fue mejor el mantener una estructura lógica, y le dimos el mismo nombre que hemos puesto anteriormente a la ruta de acceso, pero añadiendo un símbolo \$ al final del nombre, para que el recurso sea oculto, y que no se muestre si alguien realiza una exploración de la red local [18].

Path	Specify the share name to be used with the specified local path. If the share already exists on this computer, it must point to the path specified for this deployment share.
Share	
Descriptive Name	
Options	
Summary	
Progress	
Confirmation	

Share name:	<input type="text" value="MDTLab\$"/>
Full path UNC path:	\\VMDTMN\MDTLab\$

Figura 13: El nombre del recurso compartido asociado al Deployment Share.

Escribimos una descripción ampliada del punto de despliegue, en nuestro caso, simplemente repetimos el nombre del DS.

Path	Specify a descriptive name for the deployment share.
Share	
Descriptive Name	
Options	
Summary	
Progress	
Confirmation	

Deployment share description:	<input type="text" value="MDT Lab"/>
-------------------------------	--------------------------------------

Figura 14: Descripción ampliada del Deployment Share.

Se seleccionaron las opciones de despliegue de la figura 15. Esto configuró las opciones que queremos que muestre el asistente de despliegue al usuario. Estas opciones se pueden cambiar posteriormente.

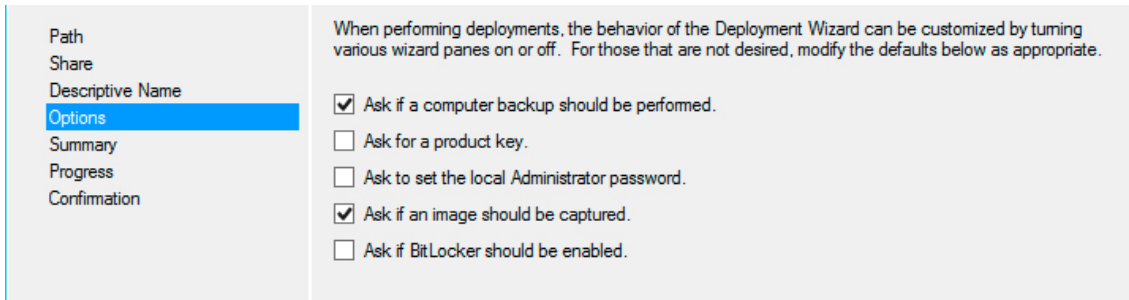


Figura 15: Las opciones de creación del Deployment Share.

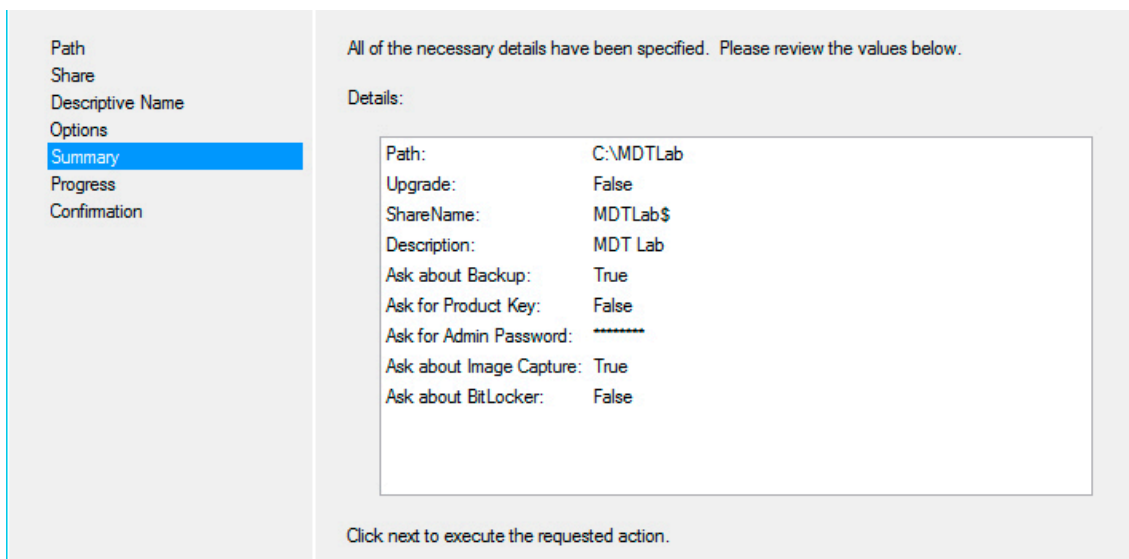


Figura 16: Resumen de la creación del Deployment Share.

Cuanto termina el proceso de creación del DS, se muestra un informe donde se nos indica que ha sido creado satisfactoriamente.

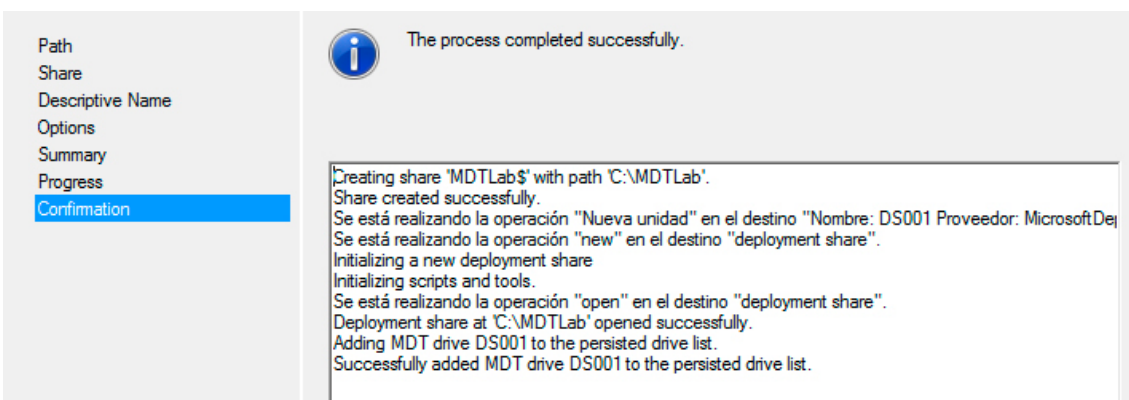


Figura 17: Mensaje de confirmación de la creación correcta del Deployment Share.

Ahora ya tenemos creado el punto de despliegue con todas las carpetas necesarias.

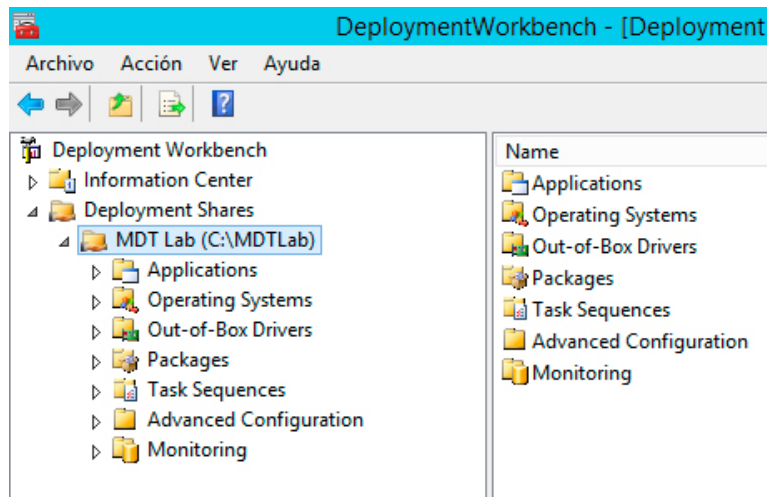


Figura 18: El Deployment Share MDTLab ya creado en el Deployment Workbench.

4.4.1.1. Carpetas que componen un DS

Cuando se crea un nuevo DS, MDT genera una estructura de carpetas vacías para cada DS creado. De este modo cada DS mantiene su propia configuración. Se ha creado un DS llamado MDTLab para hacer pruebas y más tarde se crearán un par de DS más, uno para realizar capturas de imágenes de nuestro ordenador de referencia, y otro para desplegar las imágenes en nuestro entorno de producción.

Dentro del DS MDTLab tenemos las siguientes carpetas:

- Applications. En esta carpeta añadiremos las aplicaciones que queramos instalar, bien para capturar una imagen, o bien a la hora de desplegar una imagen.
- Operating Systems. Aquí colocaremos los sistemas operativos que queramos desplegar.
- Out-of-Box Drivers. Aquí ira una estructura organizada de carpetas que contendrán los controladores de dispositivos
- Packages. Son paquetes integrados por varias aplicaciones. Podemos crear un paquete que instale dos aplicaciones, por ejemplo, que instale Office 2016 y Visio 2016
- Task Sequences. Son las secuencias de tareas que utilizaremos para capturar una imagen de un ordenador de referencia, o bien para desplegar una imagen.
- Advanced Configuration. Esta carpeta contiene la configuración avanzada de MDT. A su vez contiene otras cuatro carpetas:
 - ◊ Selection Profiles. Aquí colocaremos los perfiles de selección para filtrar los controladores que se instalaran cuando se despliega una imagen.
 - ◊ Linked Deployment Shares. Sirve para replicar el contenido de un servidor de MDT en otro, de modo que se puede balancear la carga de red cuando

se despliegan las imágenes. Se puede de varias maneras, una es usar LDS o utilizar Distributed File System Replication (DFS-R).

- ◊ Media. En esta carpeta se crean los ficheros locales para hacer un despliegue de un medio físico, como puede ser un DVD o una memoria flash.
- ◊ Database. Dentro de esta opción se configurará la base de datos que utilizara MDT para pre-almacenar los datos de los ordenadores que nos harán falta para poder realizar despliegues ZTI.
- Monitoring. Si está configurada la opción de monitorización, dentro de esta carpeta nos mostrara la actividad de los despliegues.

4.4.1.2. Actualizar el DS

Seguidamente se actualiza el punto de despliegue para que MDT cree los ficheros necesarios para poder realizar un despliegue. Se marca el punto de despliegue recién creado, en nuestro caso, MDTLab (C:\MDTLab), pulsando el botón de la derecha del ratón y marcamos sobre Update Deployment Share. Nuevamente se utilizó el asistente para ir seleccionando diversas opciones.

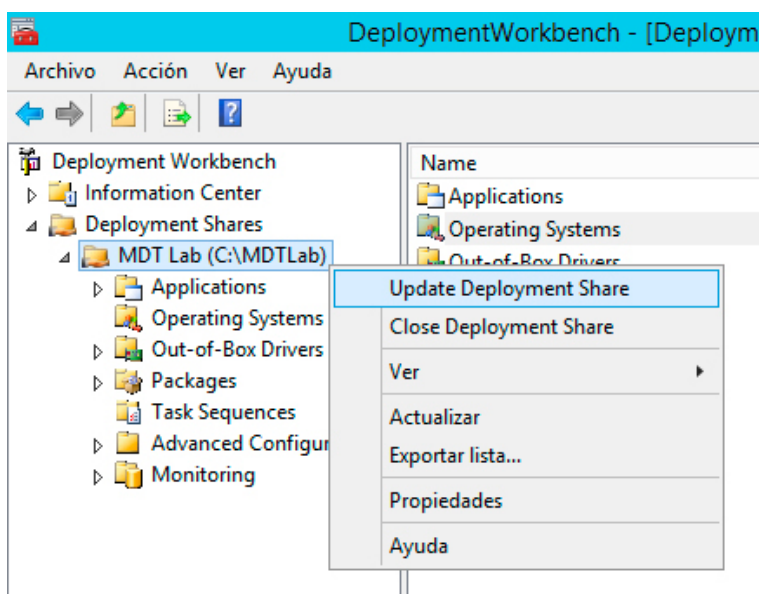


Figura 19: Actualización del Deployment Share MDTLab.

Se van a copiar las herramientas necesarias en el punto de despliegue. Hay que recordar que, si se hace algún cambio en la configuración del DS, se debe volver a realizar la actualización del punto de despliegue.

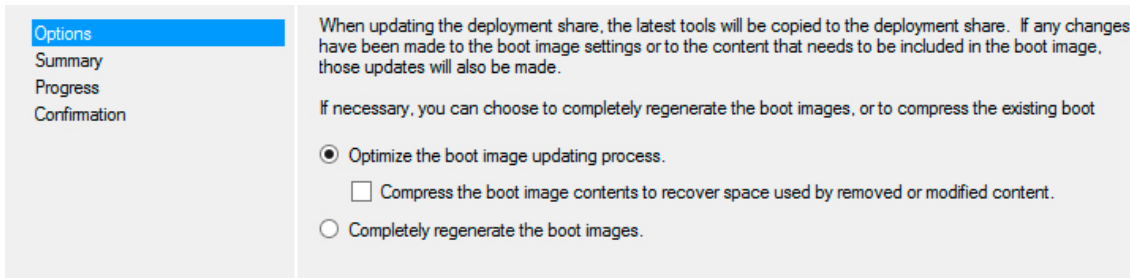


Figura 20: Opciones de actualización del Deployment Share MDTLab.

Se muestran las opciones seleccionadas y se comienza el proceso de actualización.

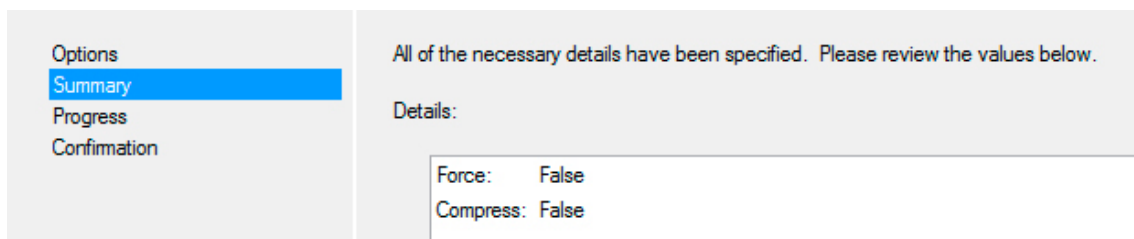


Figura 21: Resumen de la actualización del Deployment Share MDTLab.

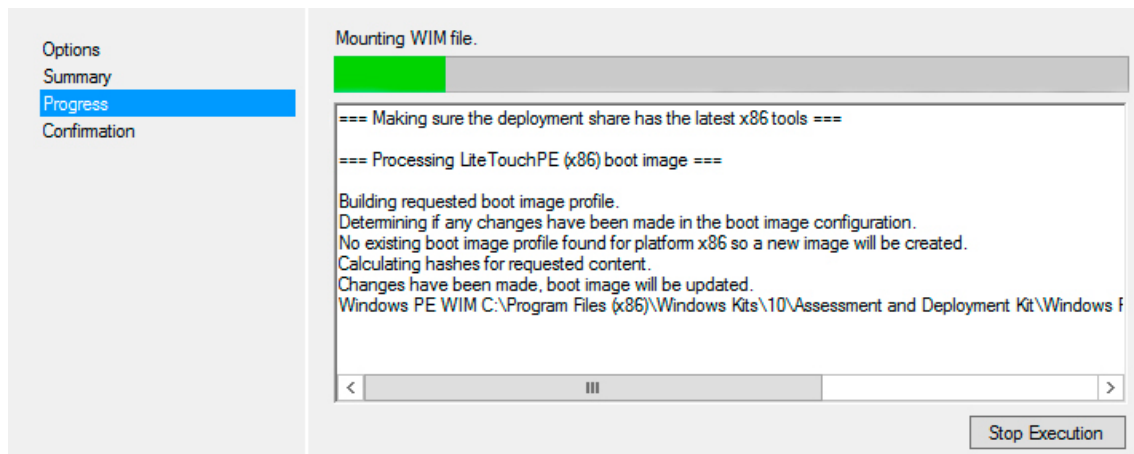


Figura 22: Progreso de la actualización del Deployment Share MDTLab.

Se finaliza la actualización del punto de despliegue, y se muestra un informe de que todo había finalizado correctamente. El proceso finaliza pulsando sobre el botón *Finish*.



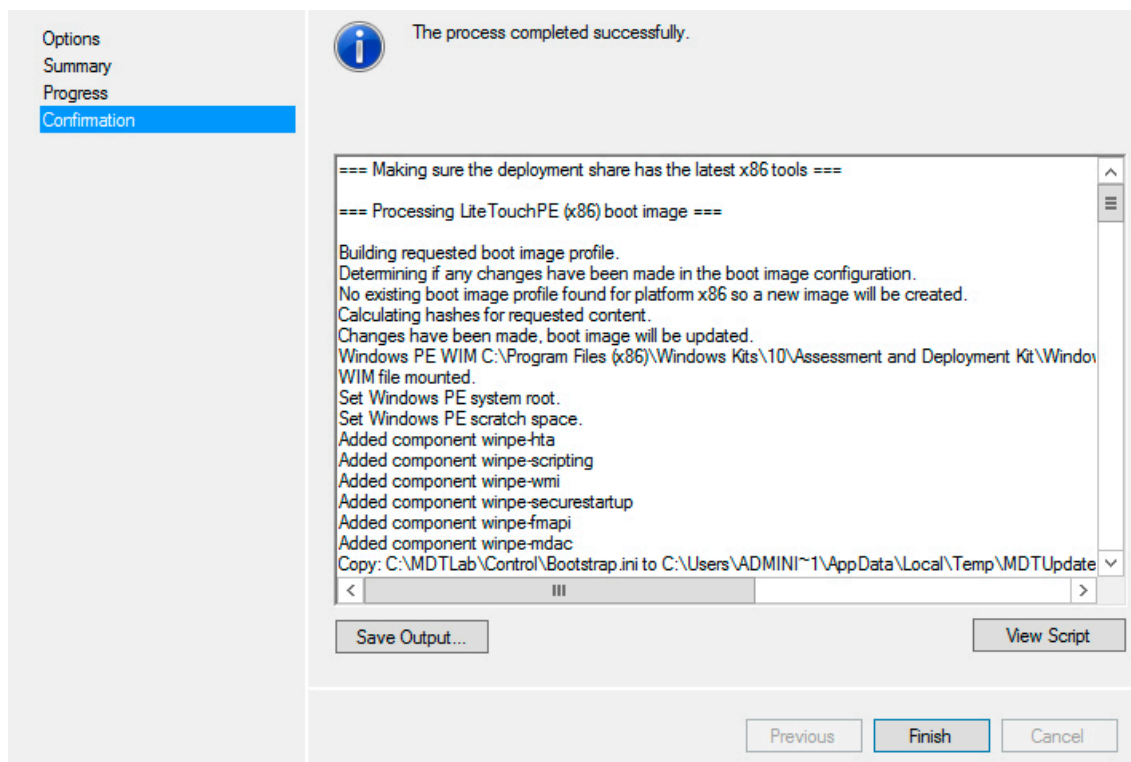


Figura 23: Informe de actualización correcta del Deployment Share MDTLab.

4.4.2. Paso número 2: crear el usuario UsuarioMDT en el servidor MDT01

Para la creación del usuario local en el servidor MDT01, se accede a la consola de administración del servidor MDT01, y se crea un usuario local, con los siguientes datos:

- Nombre de usuario: UsuarioMDT
- Contraseña: password
- Marcamos la opción: la contraseña nunca expira

Este usuario se utiliza en los despliegues desde los ordenadores cliente. Para ello, le asignamos posteriormente permisos de lectura y ejecución en el recurso compartido que hemos creado (C:\MDTLab\$) y también se le darán permisos de lectura en la base de datos que crearemos más tarde.

NOTA: La contraseña "*password*", esta puesta a modo de ejemplo. Se tiene que sustituir por la contraseña real que queramos darle al usuario.

4.4.3. Paso número 3: importar un sistema operativo.

En la primera prueba se importa un sistema operativo en la carpeta de sistemas operativos. Se marca la carpeta Operating Systems del punto de despliegue y con el botón derecho del ratón se selecciona Import Operating System. Se muestra el asistente para importar un nuevo sistema operativo al DW. Para estas pruebas utilizaremos la versión de

Windows 10 Enterprise de 64 bits. Se utilizará un DVD de la distribución de Windows 10, pero también podíamos haber usado una imagen ISO de dicho DVD. Aunque se puede importar todo el contenido del DVD en la carpeta de sistemas operativos, se va a tratar la imagen del DVD de distribución de Windows 10 como si fuera una imagen personalizada, de modo que solo se necesita el fichero WIM de instalación de Windows 10.

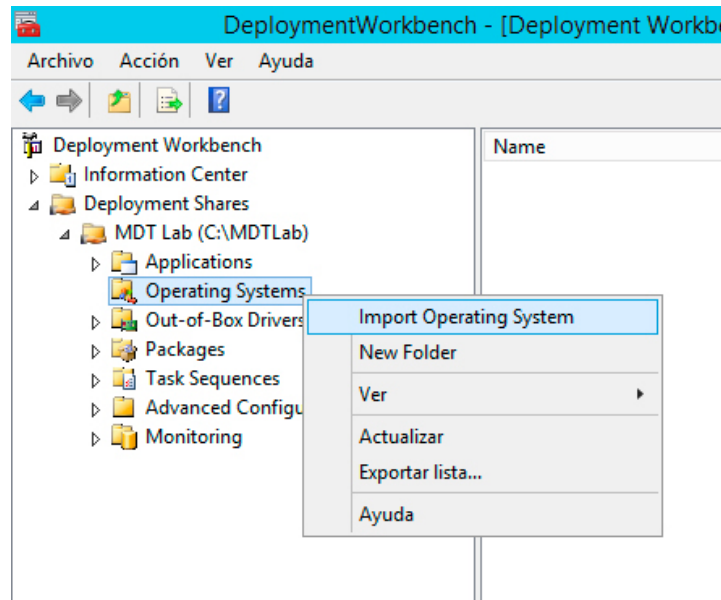


Figura 24: Como importar un nuevo sistema operativo al DS MDTLab.

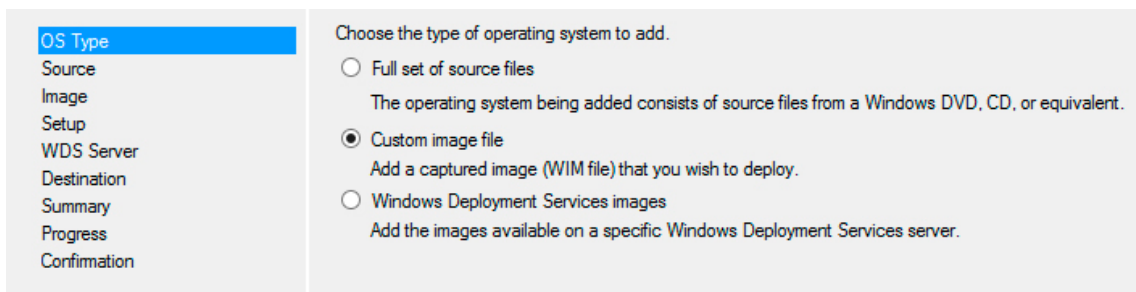


Figura 25: Opciones de Importación de un nuevo sistema operativo.

Se muestra un explorador de archivos para que se elija el fichero WIM que interesa. En el caso del DVD de Windows, el fichero de instalación se encuentra dentro de la carpeta sources y se llama install.wim; lo seleccionamos y pulsamos abrir.

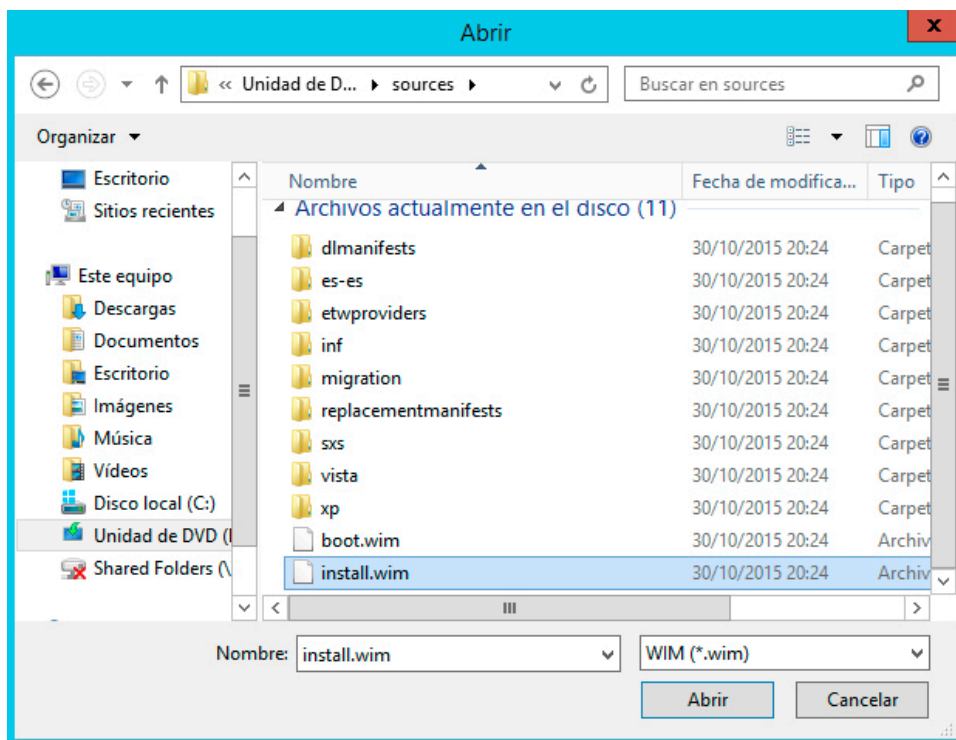


Figura 26: Explorador de archivos para importar un nuevo sistema operativo.

Se muestra la ruta al fichero WIM que se quiere importar. Existe la posibilidad de en vez de copiar el fichero a la carpeta de sistemas operativos del punto de despliegue, moverlo, pero solo si está ubicado en una carpeta donde se puede borrar.

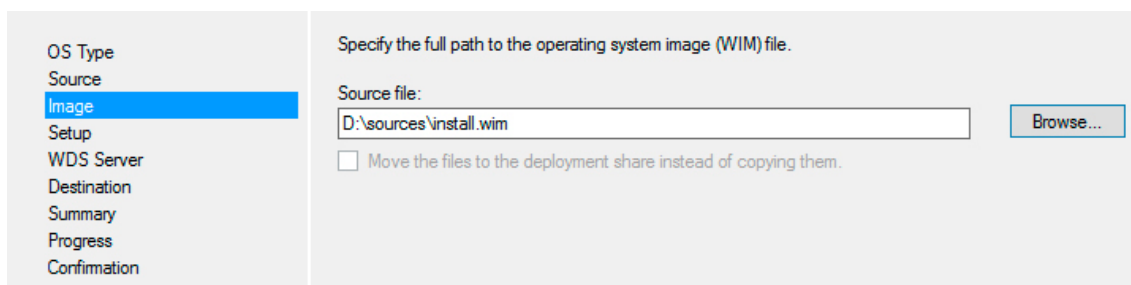


Figura 27: Ruta completa al fichero install.wim para importar.

Al tratar el fichero WIM como si fuera una imagen personalizada, no son necesarios los ficheros de instalación.

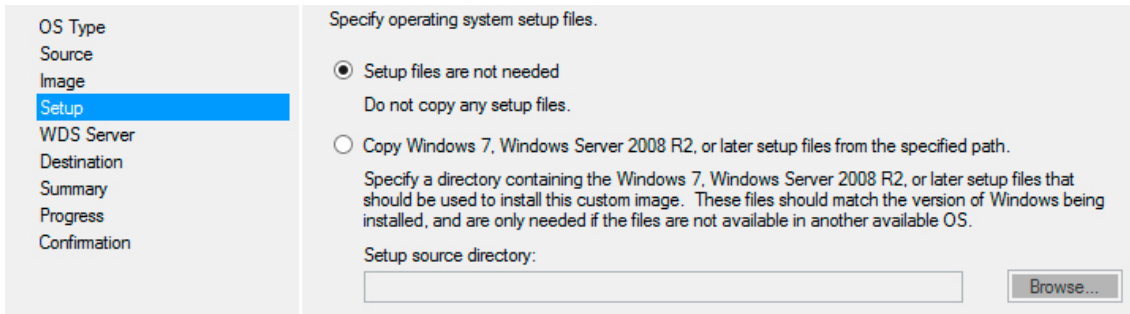


Figura 28: Opción de que no son necesarios los ficheros de instalación.

Se especificó el nombre de la carpeta donde se almacenó el sistema operativo dentro del DS.

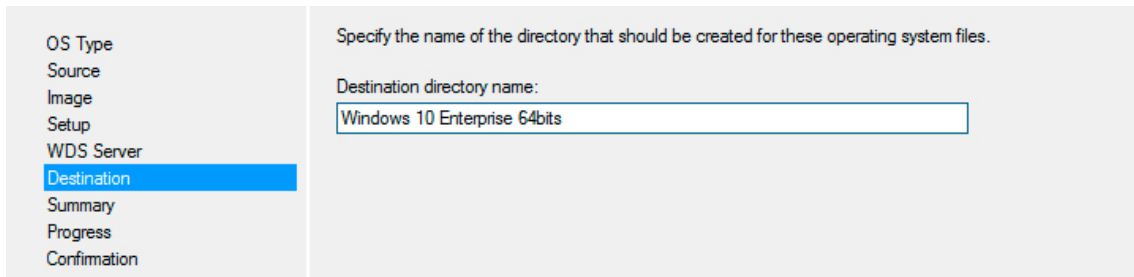


Figura 29: Nombre de la carpeta donde se almacenará el sistema operativo.

Se muestra un resumen de la configuración para este sistema operativo.

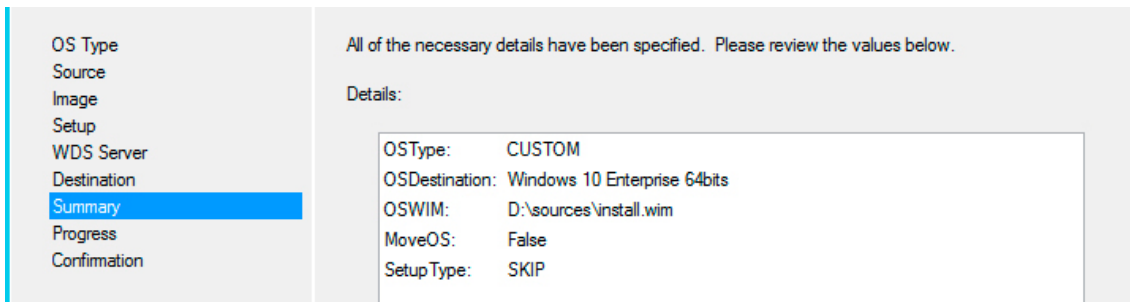


Figura 30: Resumen de la configuración para importar un sistema operativo.

Finalizada la importación se muestra un resumen del resultado de la operación.

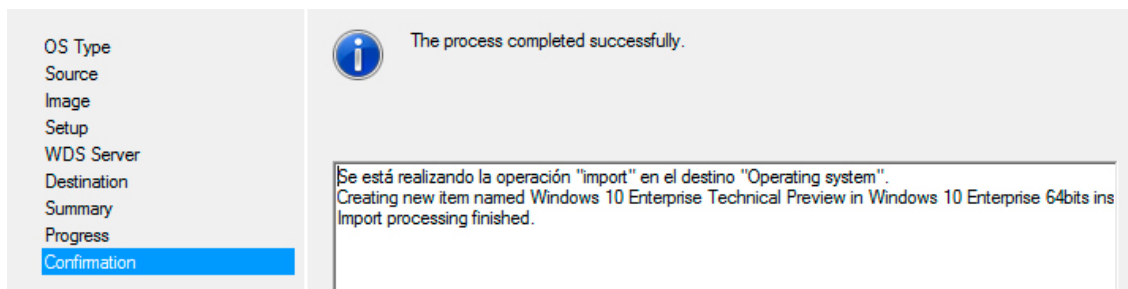


Figura 31: Mensaje de comprobación de la correcta importación del sistema operativo.

En este punto el sistema operativo ya ha sido importado en el DS, en la carpeta que hemos creado para él, dentro de la carpeta de sistemas operativos.

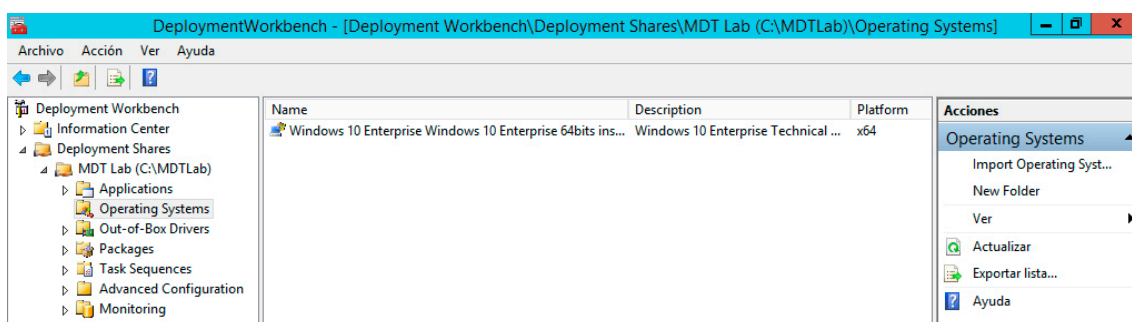


Figura 32: El sistema operativo importado en el Deployment Workbench.

4.4.4. Paso número 4: crear una secuencia de tareas

Para poder realizar el despliegue de este sistema operativo, se creó una secuencia de tareas (Task Sequence). Para ello se seleccionó Task Sequence, con el botón derecho del ratón, y se marcó New Task Sequence, mostrándose el asistente de creación de una tarea de secuencias.

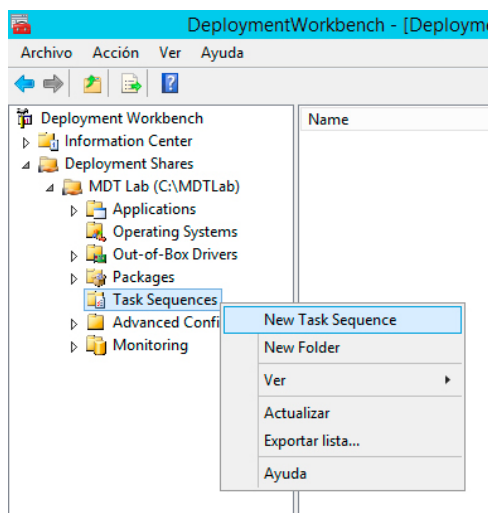


Figura 33: Como crear una nueva secuencia de tareas.

Ahora se introducen los datos que identifican a esta tarea; se escribe un identificador de tarea, que será único para todas las secuencias de este punto de despliegue. Es recomendable el escribir un identificador con pocos caracteres, pero que nos ayude a recordar de que sistema operativo se trata.

- **Task sequence ID:** W10x64ENT
- **Task sequence name:** Windows 10 Enterprise x64
- **Task sequence comments:** Despliega imagen de Windows 10 Enterprise x64 original del DVD. El objetivo es tener una descripción que nos ayude posteriormente a recordar el sistema operativo que despliega la tarea.

Specify general information about this task sequence. The task sequence ID is used internally as part of the deployment process. The task sequence name and comments are displayed by the deployment wizard.

Task sequence ID:

Task sequence name:

Task sequence comments:

Figura 34: Datos identificativos de la nueva secuencia de tareas.

De la lista de plantillas ofertadas se eligió Standard Client Task Sequence, la secuencia normal para un cliente. Hay que tener en mente que son plantillas preparadas para funcionar sin tener que modificar nada, pero pese a todo, después de crearla, se puede modificar la tarea de acuerdo a nuestras necesidades.

The following task sequence templates are available. Select the one you would like to use as a starting point.

Standard Client Task Sequence

Sysprep and Capture

Standard Client Task Sequence

Standard Client Replace Task Sequence

Standard Client Upgrade Task Sequence

Custom Task Sequence

Litetouch OEM Task Sequence

Standard Server Task Sequence

Standard Server Upgrade Task Sequence

Post OS Installation Task Sequence

Deploy to VHD Client Task Sequence

Deploy to VHD Server Task Sequence

Figura 35: Selección de la plantilla a utilizar en el despliegue.

A continuación, se elige que sistema operativo queremos desplegar. MDT nos muestra una lista con todos los sistemas operativos que hay importados en el punto de despliegue. En nuestro caso lo teníamos uno: Windows 10 Enterprise de 64 bits.

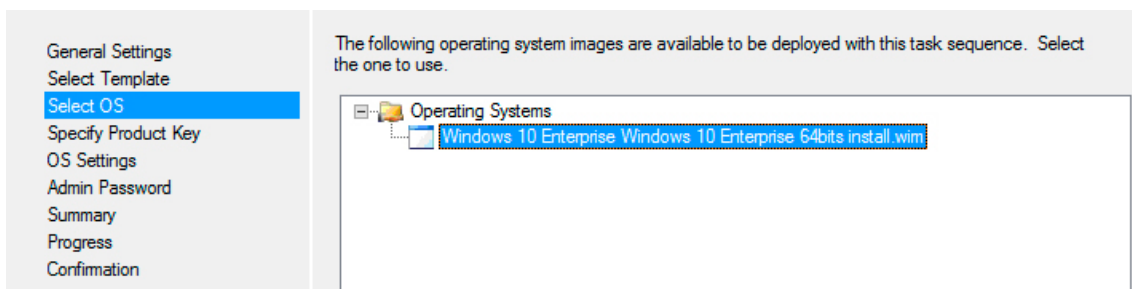


Figura 36: Se muestra el sistema operativo a desplegar.

Todos los sistemas operativos Windows requieren activarse para poder funcionar. Para conseguir esta activación se puede optar por una de estas tres opciones que nos ofrece el asistente:

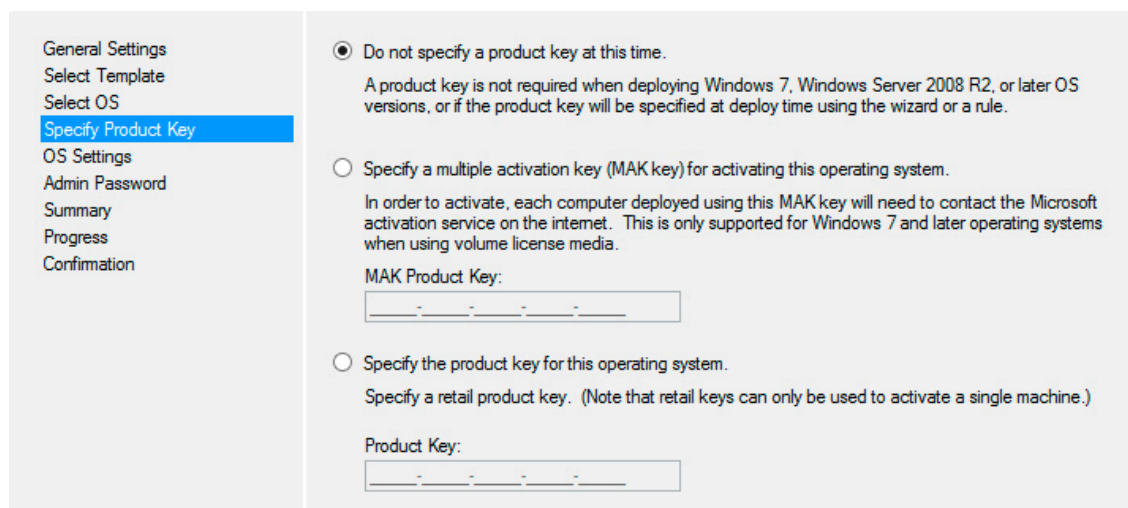


Figura 37: Opciones del tipo de activación del sistema operativo.

- Si no se especifica ninguna clave de producto, Windows tendrá que ser activado contra un servidor de licencias.
- Mediante una clave MAK de licencia. Es una clave que permite realizar múltiples activaciones del sistema operativo en diferentes ordenadores.
- Escribir una clave de producto que solo servirá para activar un único ordenador.

Nosotros seleccionamos: **Do not specify a product key at this time.** Este método es el que se usa en la UPV.

El sistema operativo guarda la información del propietario del sistema operativo y de la organización a la que pertenece. Esta información será la que se mostrará cuando el sistema operativo este instalado. También se puede especificar una página de inicio para el navegador Internet Explorer. Completamos el asistente con los siguientes datos:

- **Full Name:** ETSIAMN
- **Organization:** UPV

- **Internet Explorer Home Page:** www.upv.es

<ul style="list-style-type: none"> General Settings Select Template Select OS Specify Product Key OS Settings Admin Password Summary Progress Confirmation 	<p>Specify settings about this task sequence. These settings will be used for all deployments of this task sequence, unless overridden during the deployment process using the wizard or a rule.</p> <p>Full Name: <input type="text" value="ETSIAMN"/></p> <p>Organization: <input type="text" value="UPV"/></p> <p>Internet Explorer Home Page: <input type="text" value="www.upv.es"/></p>
--	--

Figura 38: Información relativa al propietario del sistema operativo y de la página de inicio de Internet Explorer.

Hubo que especificar la contraseña del usuario administrador local del ordenador. El usuario administrador está siempre creado por defecto en el sistema operativo Windows, pero se encuentra en estado desactivado. MDT utiliza la cuenta de administrador para realizar el despliegue y hacer tareas posteriores con privilegios elevados, una vez desplegada la imagen. MDT se encarga de activar la cuenta. Si queremos, después de desplegar la imagen, y si no vamos a utilizar la cuenta de administrador local, podemos deshabilitarla de nuevo. Si no especificamos una contraseña ahora, durante el despliegue nos solicitará que escribamos una.

<ul style="list-style-type: none"> General Settings Select Template Select OS Specify Product Key OS Settings Admin Password Summary Progress Confirmation 	<p>Specify the local Administrator password for this task sequence.</p> <p><input checked="" type="radio"/> Use the specified local Administrator password.</p> <p>Administrator Password: <input type="password" value="●●●●●●"/></p> <p>Please confirm Administrator Password: <input type="password" value="●●●●●●"/></p> <p><input type="radio"/> Do not specify an Administrator password at this time.</p> <p>The local Administrator password will be provided during the deployment of this task sequence, so it is not needed as part of the task sequence definition.</p>
--	---

Figura 39: Opciones donde se puede introducir, o no, la contraseña del usuario Administrador.

Al concluir este proceso se muestra un resumen de las opciones de la creación de la tarea y la comprobación de que la tarea se había creado correctamente.

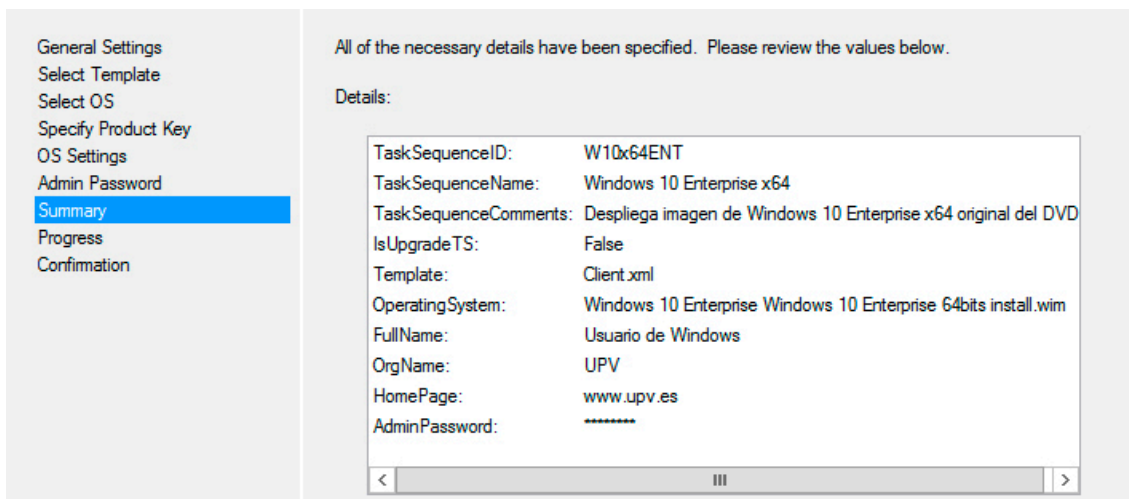


Figura 40: Resumen de la creación de la tarea de secuencias.

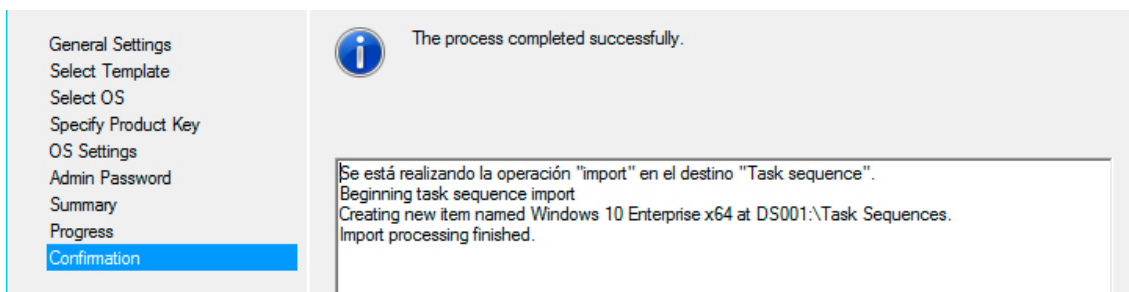


Figura 41: Informe una vez finalizado el proceso de creación de la tarea de secuencias.

Cuando termina el asistente, la secuencia de tareas ya ha sido incorporada a la carpeta de secuencias de tareas.

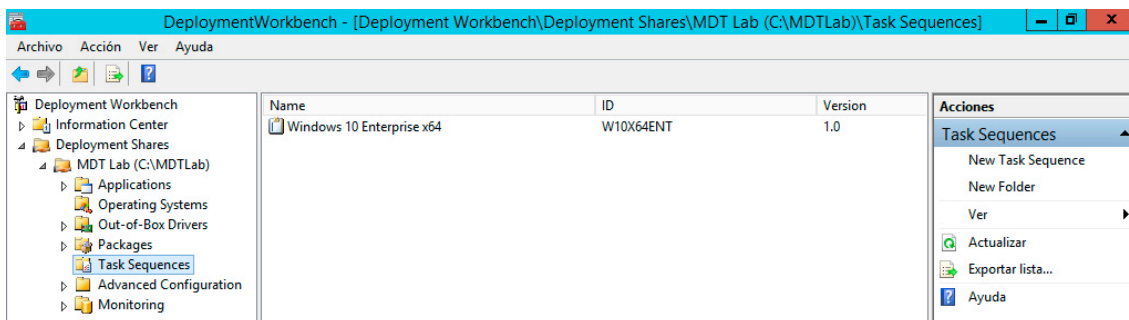


Figura 42: La tarea recién creada en el Deployment Workbench.

4.4.5. Paso número 5: desplegar la imagen de pruebas

Se realiza un despliegue de pruebas con la tarea recién creada y con el sistema operativo importado. Para ello se debe de arrancar el ordenador utilizando un sistema operativo basado en Windows PE y que este almacenado en un medio físico externo, como puede ser un disco DVD o una memoria *flash* USB. Cuando actualizamos el punto

de despliegue en el punto 4.4.1.2., se crearon dentro de la carpeta Boot las imágenes de arranque para ese DS. En nuestro caso la imagen de arranque de 64 bits es el fichero LiteTouchPE_x64.iso, que es el que tenemos que grabar en el disco DVD o en una memoria flash USB. Una vez arranca el ordenador por medios físicos externos se muestra el asistente para hacer el despliegue; que da la opción a comenzar o a salir del despliegue. Si elegimos la opción de salir, se muestra una ventana de símbolo de comandos y finaliza el proceso de despliegue. Otras opciones son la selección del idioma de nuestro teclado y configurar una IP estática para el ordenador.

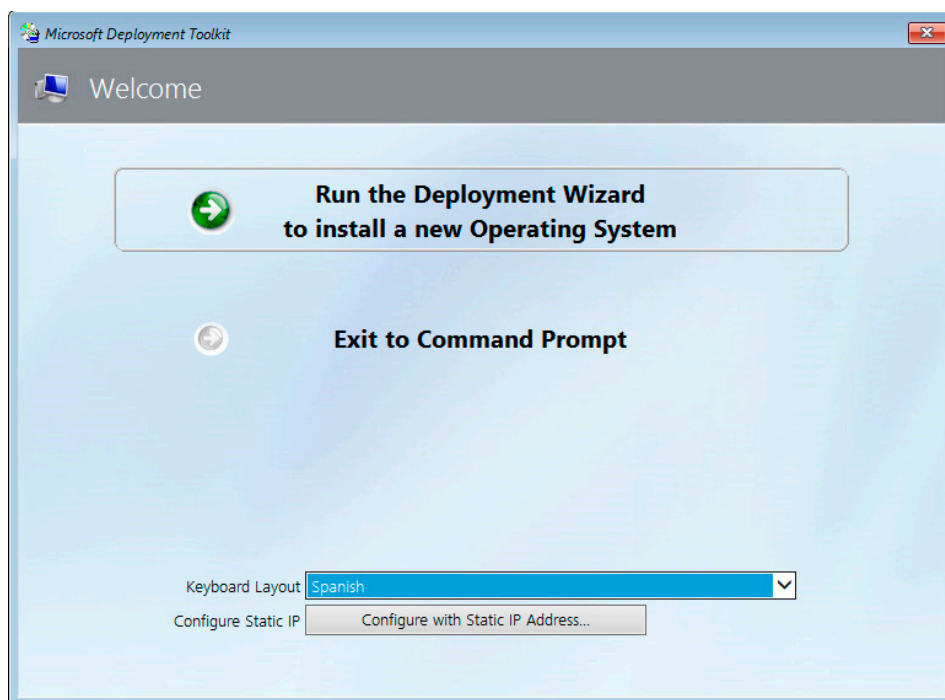


Figura 43: Asistente de despliegue LTI.

Se debe especificar el servidor y el punto de despliegue que se va a utilizar. En este caso, el nombre del servidor fue MDT01 y el del punto de despliegue fue MDTLab\$ (\\MDT01\MDTLab\$).

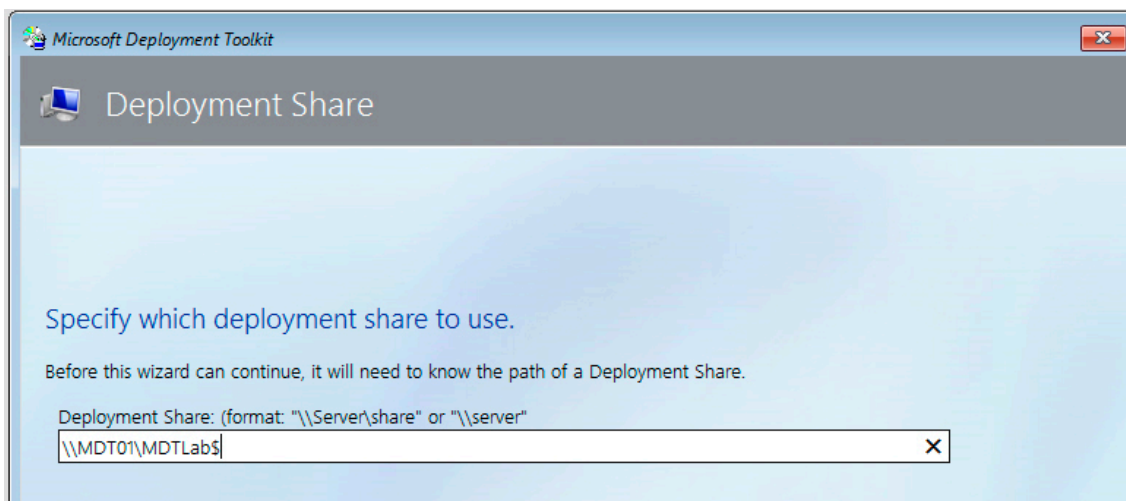


Figura 44: Especificación del servidor y el punto despliegue.

Como se está realizando un despliegue LTI, se solicitan las credenciales para conectarse con el punto de despliegue. Se introducen los datos del usuario que creamos en el punto 4.4.2.:

- **User Name o nombre usuario:** UsuarioMDT
- **Password o contraseña:** password
- **Domain o dominio:** MDT01. El dominio del ordenador en este caso es el propio nombre del servidor y no un dominio.

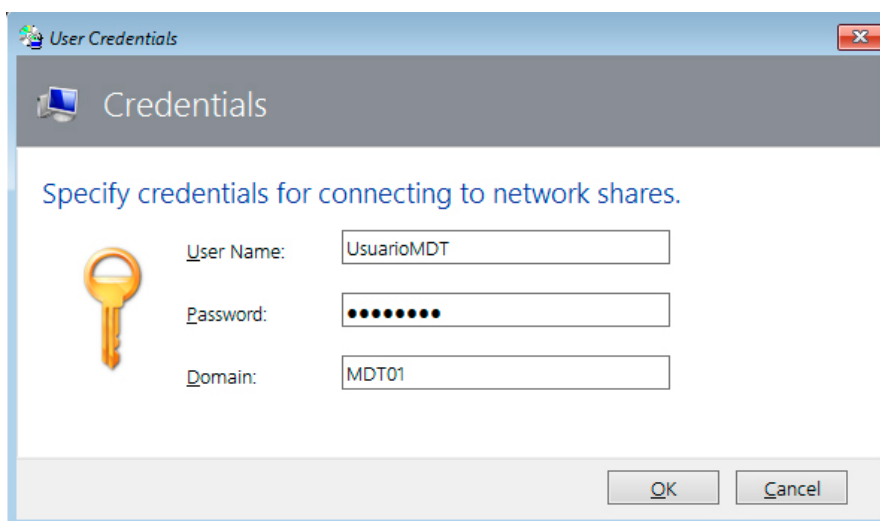


Figura 45: Credenciales para conectar con el punto despliegue.

Una vez validadas las credenciales del usuario, se muestra una lista con todas las secuencias que hay en el punto de despliegue. En nuestro caso solo muestra una, y se selecciona.

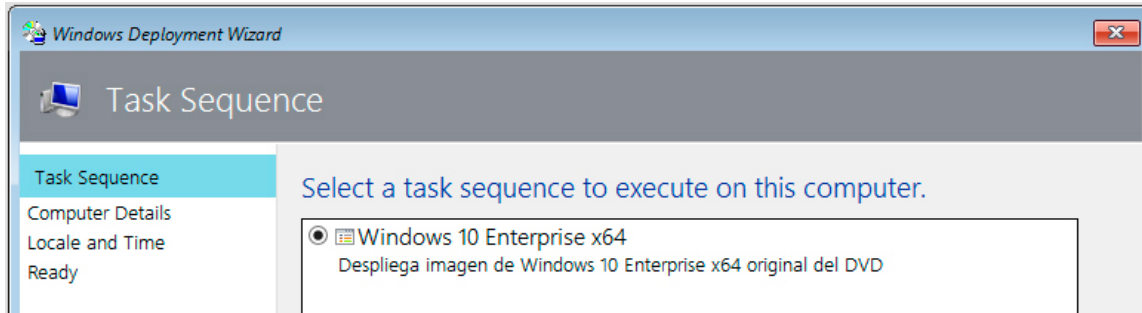


Figura 46: Listado de secuencias de tareas a desplegar.

En Computer Details hay que introducir los datos referentes al ordenador:

- **Computer name:** PruebaW10
- Seleccionamos **Join a domain**
 - **Domain to join:** alumno
 - **User Name:** agregar
 - **Password:** password
 - **Domain:** alumno

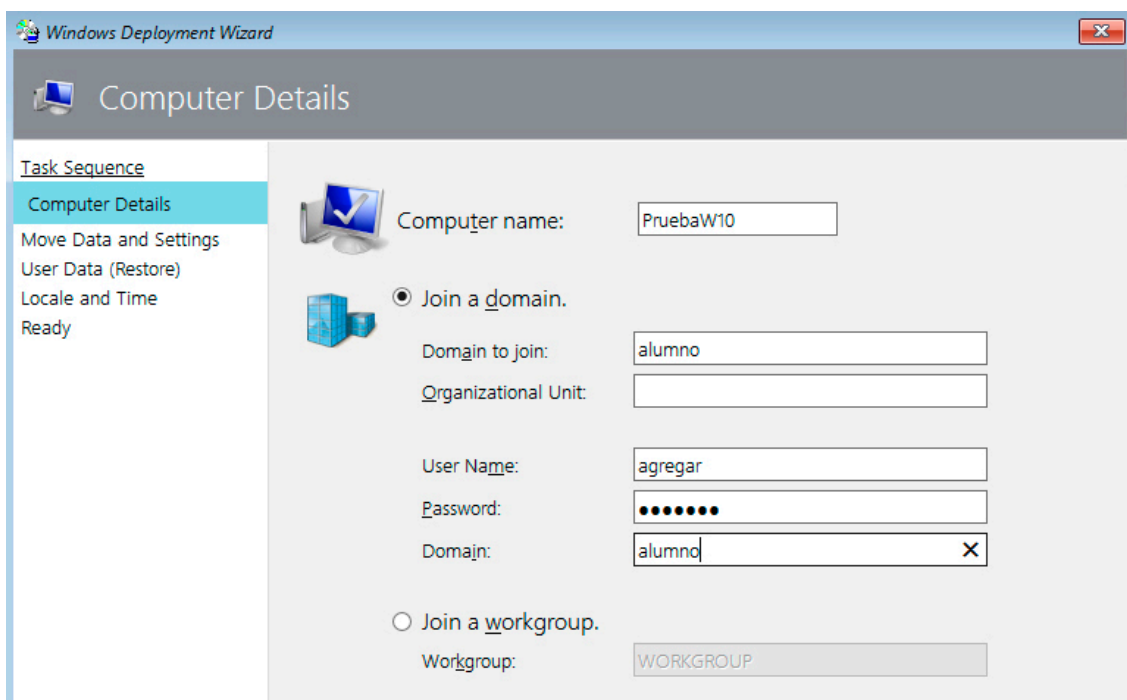


Figura 47: Datos referentes al ordenador.

En los dos siguientes pasos nos solicita si queremos guardar y posteriormente recuperar, una copia de seguridad de los datos de los usuarios del ordenador. Los ordenadores de las aulas de informática de la ETSIAMN, los utilizan a diario numerosos usuarios, que deben de realizar copias de sus datos y documentos en medios de almacenamiento externos al ordenador. Por este motivo se seleccionó en el asistente las opciones de no guardar una copia de seguridad de los datos de usuario, y de no recuperar dicha copia.

En Move Data and Settings se seleccionó la opción: **Do not move user data and settings**.

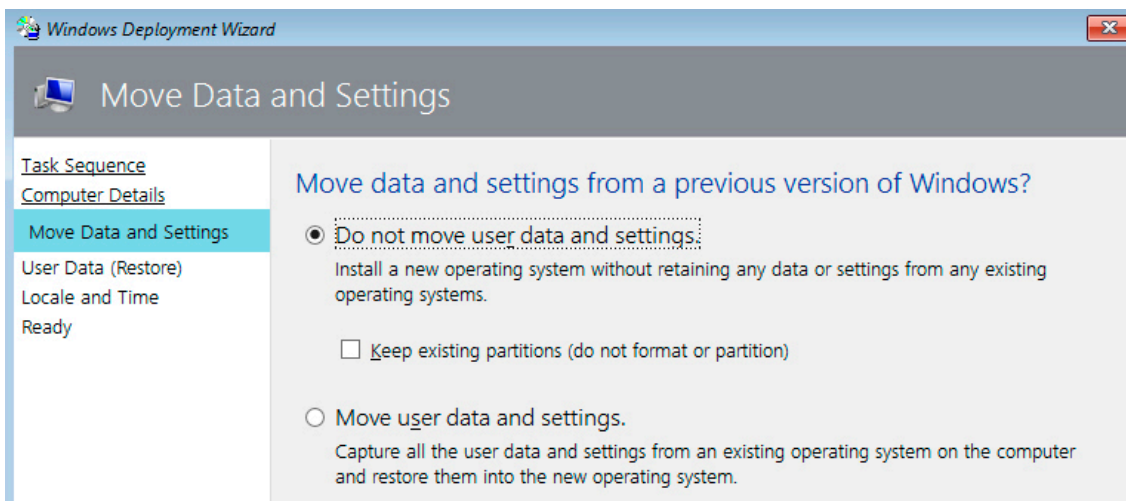


Figura 48: Selección de no mover los datos del usuario.

En User Data (Restore) se seleccionó la opción: **Do not restore user data and settings**.

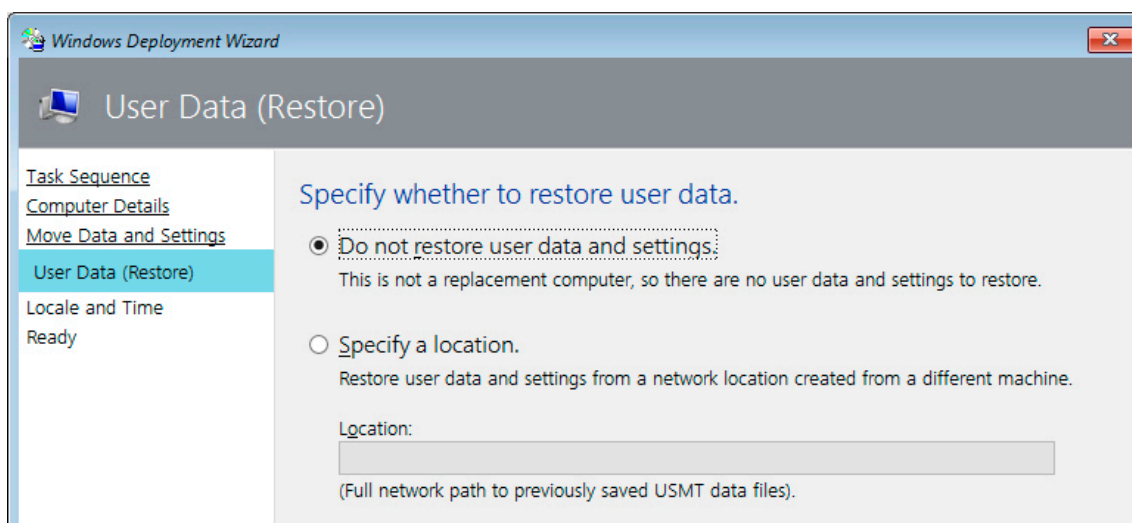


Figura 49: Selección de no restaurar los datos del usuario.

Se especificó el idioma local del ordenador, el del teclado y el formato de fecha y hora, seleccionando lo siguiente:

- **Language to install:** Spanish (Spain, International Sort)
- **Time and currency format (Locale):** Spanish (Spain, International Sort)
- **Keyboard layout:** Spanish
- **Time Zone:** (UTC +01:00) Brussels, Copenhagen, Madrid, Paris

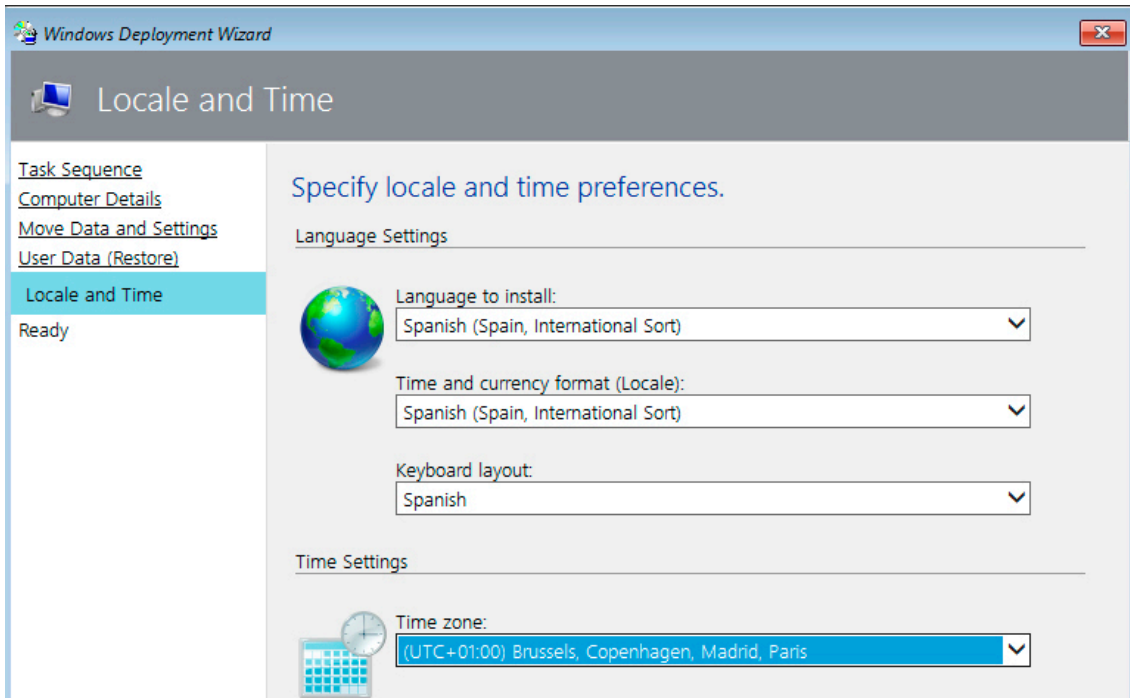


Figura 50: Configuración local de idioma y hora.

Una vez completada la configuración, el asistente nos muestra un resumen de las operaciones que MDT va a realizar en el despliegue utilizando nuestra secuencia. En este punto es importante que se repasen los datos que aparecen en el resumen para comprobar que no se han producido errores.

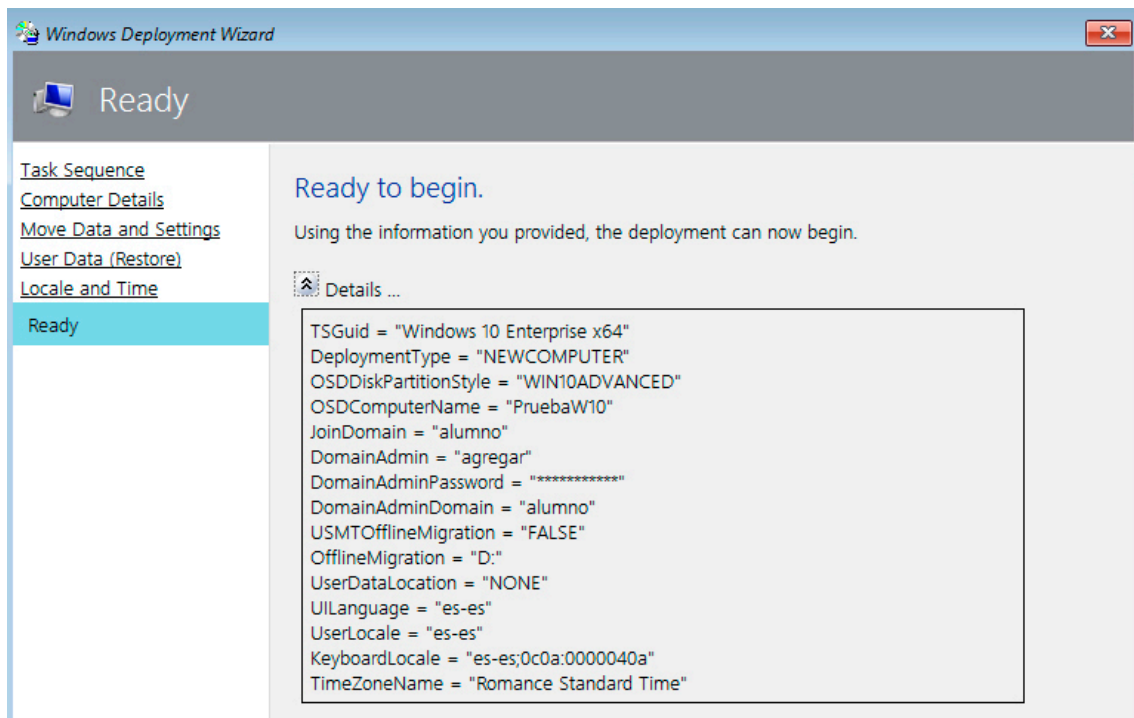


Figura 51: Resumen de la configuración del despliegue.

Ya han finalizado las operaciones previas al despliegue de imagen. MDT ha ido mostrando las acciones que iba realizando. Ahora en esta fase, MDT está recopilando la información referente al ordenador, necesaria para el despliegue. Después de unos momentos comenzará realmente el despliegue de la imagen sobre el ordenador.

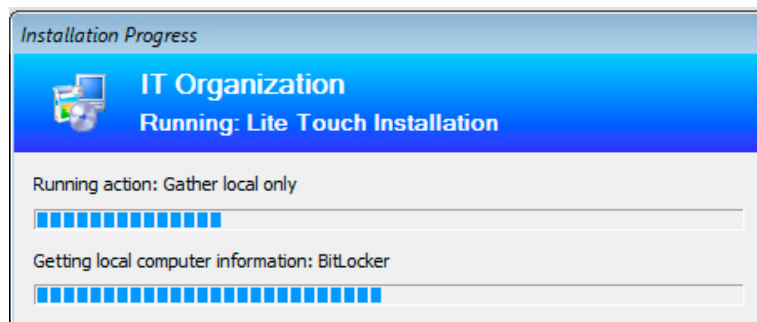


Figura 52: Recopilar información para el despliegue.

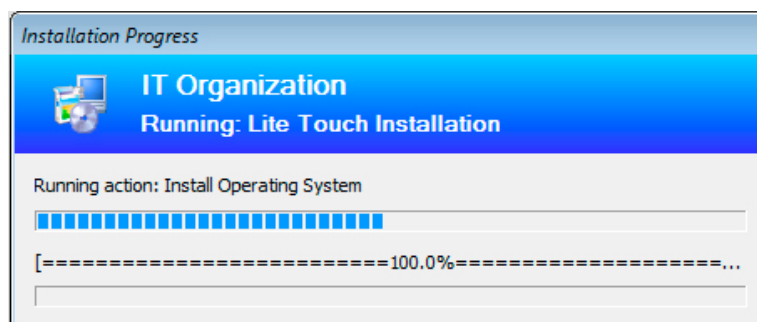


Figura 53: Barra de progreso del despliegue de la imagen.

Cuando finaliza el despliegue de la imagen, Windows arranca por primera vez y comienza la búsqueda de los controladores de dispositivos del ordenador. Una vez finalizado totalmente el despliegue, se nos mostrara un informe con el resultado del mismo. En este caso se mostró que no ocurrió ningún error y ninguna advertencia. El despliegue finalizó correctamente. En este punto el ordenador esta en el siguiente estado:

- con el usuario administrador, protegido con una contraseña
- integrado en el dominio alumno
- con una configuración por defecto.

Se pueden realizar tareas complementarias como añadir más usuarios, cambiar configuraciones, fondo de pantalla o instalar más aplicaciones.

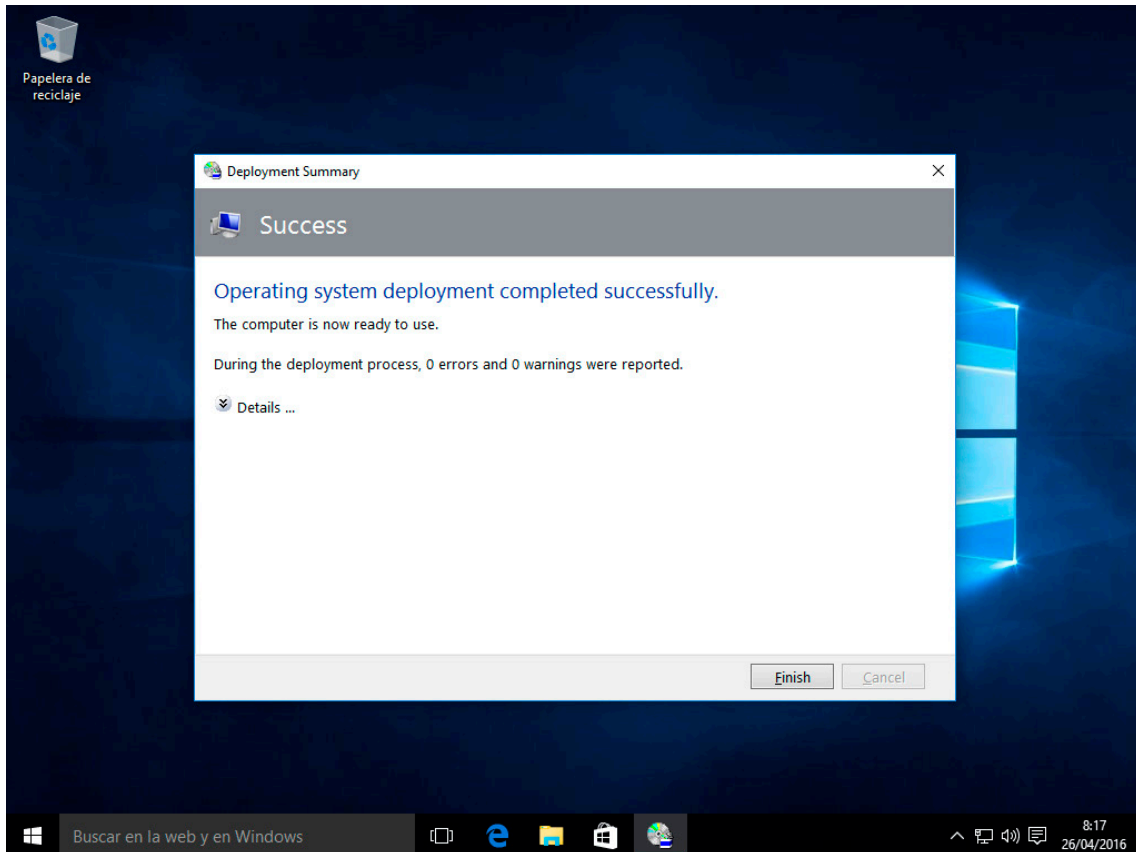


Figura 54: Mensaje de que el sistema operativo ha sido desplegado sin errores.

4.5. Crear una imagen de referencia

En el apartado anterior se importó una imagen del sistema operativo Windows 10 x64 Enterprise a partir del DVD de distribución, y la tratamos como si fuera una imagen personalizada. Ahora se creará una imagen personalizada a partir de un ordenador de referencia, que también se tuvo que preparar.

4.5.1. La plataforma de virtualización de Microsoft: Hyper-V

Para poder crear las imágenes que posteriormente desplegaremos a los ordenadores cliente, necesitamos un ordenador de referencia que contenga el menor número de controladores de dispositivos. ¿Se podría utilizar un ordenador físico? La respuesta es sí, pero está totalmente desaconsejado. La imagen de referencia no debe contener ningún controlador específico dependiente del hardware del ordenador. Ese entorno solo se puede encontrar en una máquina virtual. Para realizar esta prueba, se utilizará la plataforma de virtualización de Microsoft en Windows 2012 Server, que se llama Hyper-V [19].

Las ventajas de crear nuestro ordenador de referencia en el entorno Hyper-V son:

- Permite crear puntos de control del estado del ordenador en un momento dado, para poder devolver al ordenador a ese estado rápidamente. Esto es muy útil cuando se están realizando pruebas de implementación de aplicaciones o configuraciones, y si algo va mal, se puede regresar a un estado previo sin errores.
- Se puede realizar una copia de la máquina virtual y llevarla a otro servidor para ejecutarla allí.
- Su administración es sencilla gracias al control centralizado de todas las operaciones que se pueden realizar con las máquinas virtuales.
- Permite la simulación de una variedad de hardware. Por ejemplo:
 - ◇ Crear un ordenador con uno o varios procesadores que soporten una arquitectura de 32 o 64 bits.
 - ◇ Ajustar la cantidad de memoria RAM a nuestras necesidades.
 - ◇ Crear un disco virtual del tamaño que queramos, o bien utilizar uno o varios discos virtuales ya creados.
 - ◇ Utilizar una unidad de DVD virtual para acceder a ficheros ISO.
- Es escalable. Podemos alterar y mejorar las características del hardware si fuera necesario, como por ejemplo, añadir tarjetas de red de mayor velocidad.

4.5.2. Instalar Hyper-V en Windows Server 2012 R2

Hyper-V es un rol que no está instalado por defecto en Windows Server 2012 R2. Para su instalación se utilizó la opción número dos, agregar roles y características del panel de administración del servidor.



Figura 55: El administrador de servidor de Windows Server 2012 R2.

Una vez seleccionado, en el asistente para agregar roles y características se seleccionaron las siguientes opciones:

1. Tipo de instalación: Instalación basada en características o en roles.
2. Seleccionar servidor de destino: se eligió el servidor de nuestro caso (MDT01).
3. Seleccionar roles de servidor: marcamos Hyper-V.
4. Seleccionamos Agregar características.
5. Crear conmutadores virtuales: se creó un conmutador virtual que permitió a la máquina virtual conectarse con la red local. Se seleccionó Ethernet, que está asociado con la tarjeta de red del servidor MDT01. Ver Figura 56.
6. Migración de máquinas virtuales. No se seleccionó nada.
7. Almacenes predeterminados. No se modificó nada.
8. Se inicia la instalación pulsando en Instalar y una vez finalizada la instalación, se indica que se debe reiniciar el servidor.

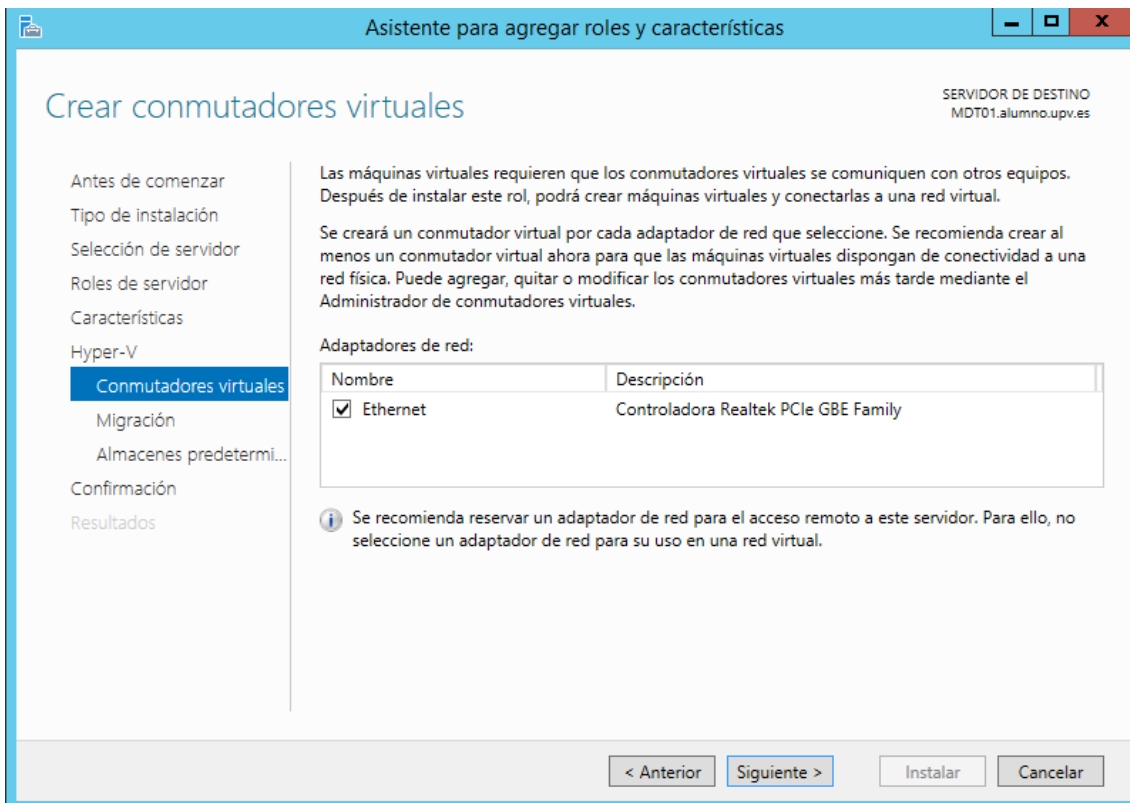


Figura 56: La creación de un conmutador virtual para Hyper-V.

4.5.3. Crear un ordenador de referencia en Hyper-V

Después de reiniciar el ordenador se ejecuta el administrador de Hyper-V, y ya se puede crear un ordenador virtual para nuestra imagen de referencia. Para ello se selecciona pulsando el botón de la derecha del ratón en el servidor MDT01, se selecciona Nuevo, y Máquina virtual.

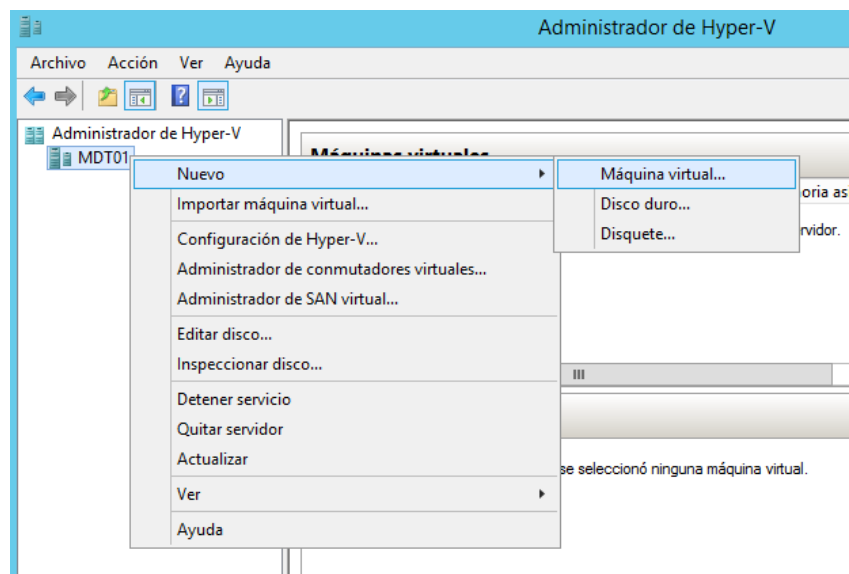


Figura 57: Como crear la máquina virtual de referencia.

A continuación, se muestra el asistente de creación de una nueva máquina virtual y se solicitan los siguientes datos:

1. Nombre: Ordenador de referencia
2. Generación: Generación 1
3. Memoria de inicio: 2048 MB
4. Conexión: Controladora Realtek PCIe GBE Family – Virtual Switch
5. Se selecciona: crear un disco duro virtual, y se deja el resto de datos por defecto
6. Se selecciona la opción: Instalar un sistema operativo más adelante

Notas:

1. Especificar generación. Se puede elegir entre dos generaciones de máquina virtual. La generación 2 tiene unas mejoras importantes respecto de la generación 1:
 - arranque seguro
 - arranque SCSI
 - arranque PXE vía red
 - soporta sistema operativo Windows 8 o superior.

Realmente no importa de qué generación es la máquina virtual, pues únicamente la necesitamos para poder leer la imagen de referencia.

2. La cantidad de memoria asignada se tendrá que ponderar en función de la memoria del ordenador donde está instalada la plataforma Hyper-V.

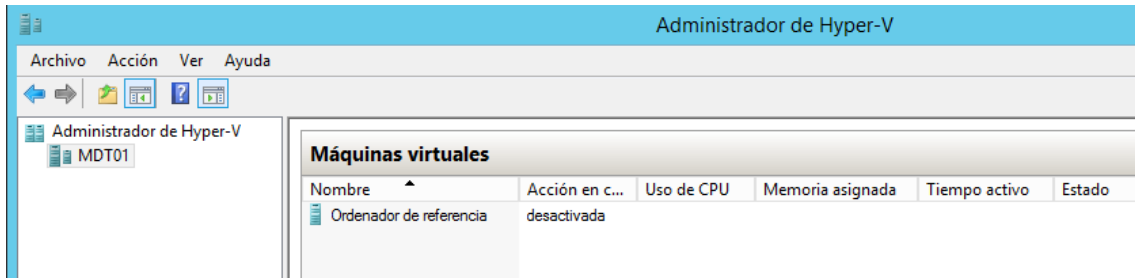


Figura 58: Máquina virtual de referencia creada en el administrador de Hyper-V.

Los únicos parámetros que se han asignado son la memoria y el tamaño del disco. Se debe revisar y/o configurar el resto de la configuración de la máquina virtual. Para ello se selecciona la máquina virtual y pulsando el botón de la derecha del ratón y se elige la opción de Configuración. Aquí se muestra toda la configuración de la máquina virtual. Se observa que el asistente ha creado la máquina virtual con un solo procesador virtual; para que la máquina virtual sea más rápida se le añade un segundo procesador virtual (pulsando sobre Procesador, se añade un procesador virtual). Se revisa el resto de la configuración, pero para nuestro objetivo de crear el ordenador de referencia, la configuración ya es correcta.

4.5.3.1. Instalación del sistema operativo en el ordenador de referencia

Primero se debe conectar con la máquina virtual (o pulsando dos veces con el botón izquierdo del ratón sobre ella, o pulsando el botón de la derecha y Conectar). Este proceso solo se conecta con la máquina virtual, no la arranca.

Antes de iniciar la máquina virtual de referencia, y utilizando el DVD virtual, se inserta una imagen ISO del sistema operativo para instalarlo. Se pulsa en el menú de Medios, Unidad de DVD, Insertar disco..., y en el explorador de archivos buscamos la imagen ISO del sistema operativo Windows 10 x64 Enterprise. Cuando iniciamos la máquina virtual, como no tiene ningún sistema operativo instalado, comenzará el proceso de instalación desde el disco del DVD virtual, y cuando finalice la instalación del sistema operativo, la máquina virtual se reiniciará para presentarnos la pantalla de configuración inicial.

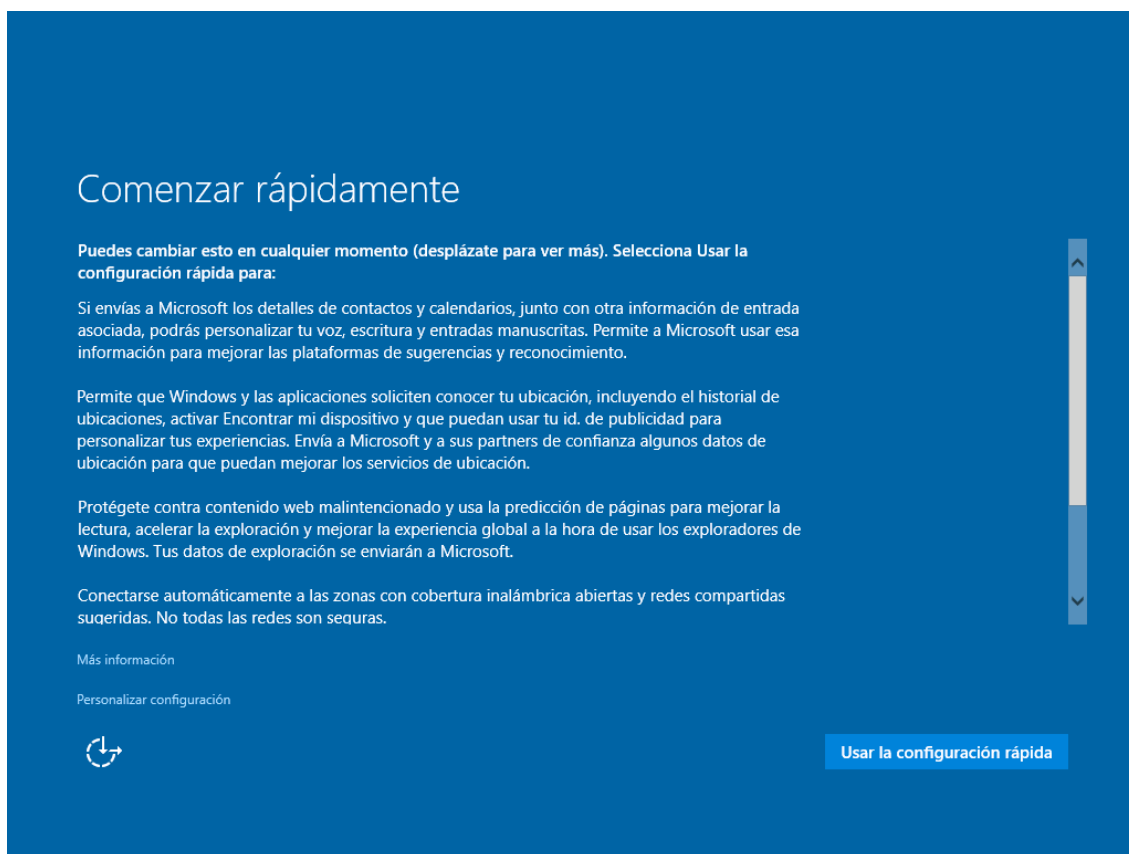


Figura 59: Asistente para la configuración inicial del sistema operativo.

4.5.3.2. Modo de auditoría

El modo auditoría permite a los OEM y a las organizaciones personalizar una instalación de Windows antes de enviar el equipo a un usuario final. En modo de auditoría, se pueden instalar aplicaciones, agregar controladores de dispositivo, ejecutar *scripts* y probar la validez de una instalación de Windows. El modo auditoría es un entorno preparado para redes que no necesita que se aplique ninguna configuración en la bienvenida de Windows [20].

Para entrar en el modo de auditoría es necesario pulsar las teclas **[Mayúsculas] + [Control] + [F3]** simultáneamente, en la asistente que se muestra en la Figura 59. Windows se reinicia y entra en el modo de auditoría.

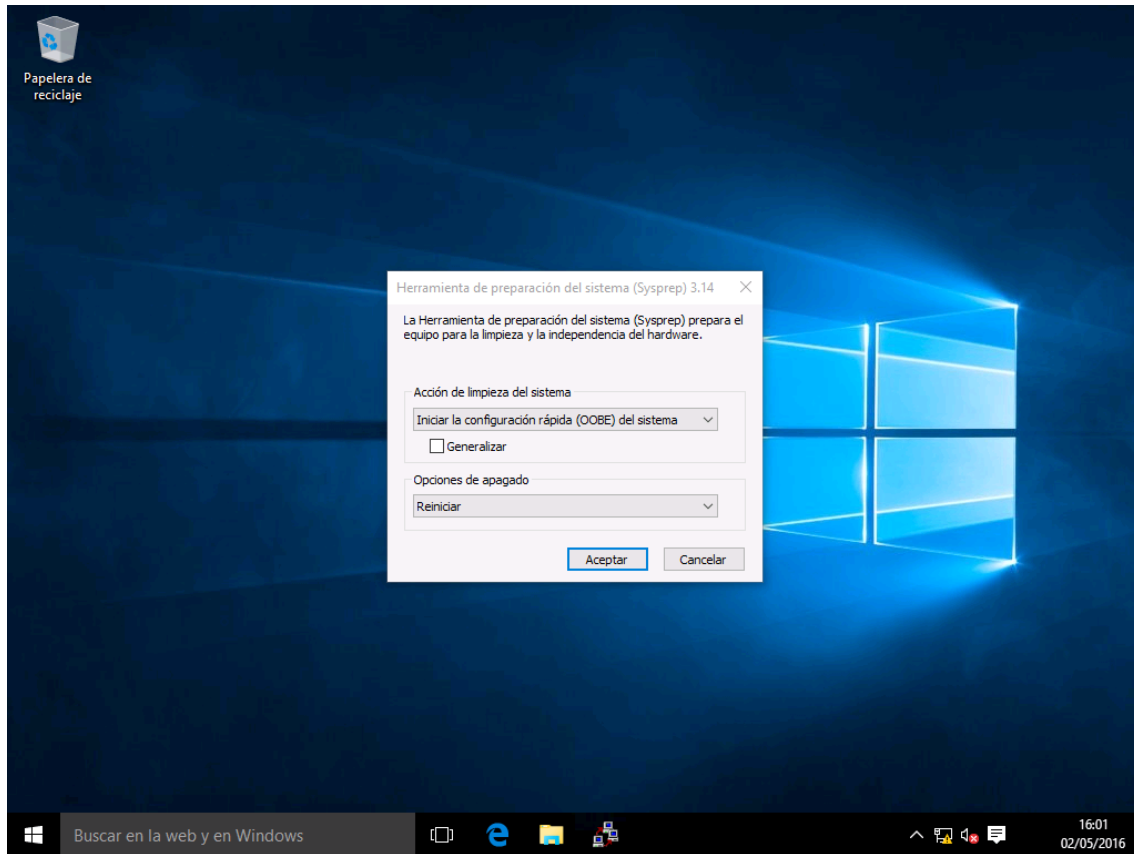


Figura 60: El sistema operativo en el modo de auditoria.

4.5.3.3. Instalar Office 2016

Se utiliza el modo de auditoria para crear nuestra imagen personalizada de referencia. En el modo de auditoria, cada vez que se reinicia la máquina virtual, aparecerá el cuadro de dialogo de la herramienta de preparación del sistema. Se debe pulsar en el botón Cancelar, para acceder al escritorio de Windows, y comenzar la instalación de aplicaciones. En el ordenador de referencia se instaló una distribución de Office 2016, para ello se utilizó una copia del programa en soporte físico DVD. También se podía haber usado una imagen ISO del DVD.

4.5.3.4. Actualizar el sistema operativo y Office 2016

Después de la instalación de Office 2016, se hizo una búsqueda de actualizaciones en WindowsUpdate, para asegurarnos que tanto el sistema operativo Windows 10 y Office 2016 contaban con los últimas actualizaciones y parches de seguridad en nuestra imagen de referencia.

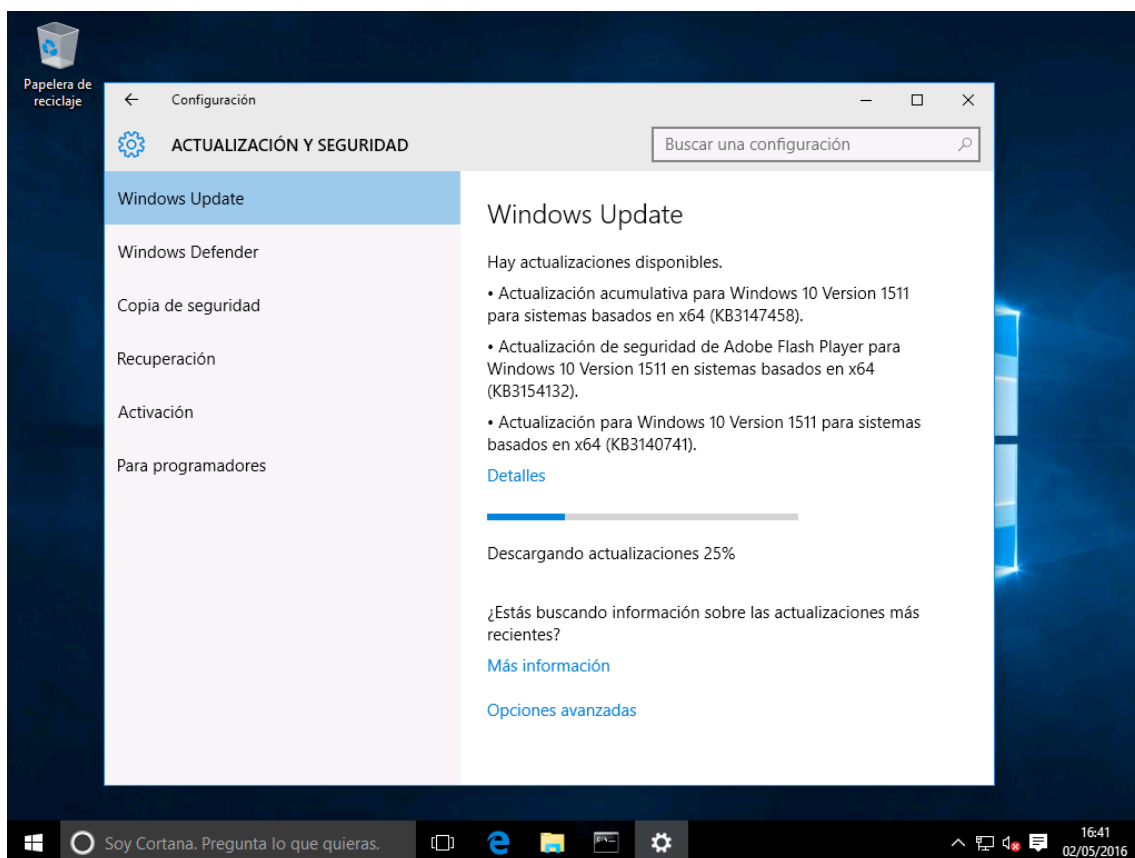


Figura 61: Actualización de Windows y Office 2016 desde WindowsUpdate.

4.5.3.5. Personalizar el escritorio

Cuando se finaliza el proceso de actualización, se procede a personalizar el escritorio; dejando en el escritorio varios iconos de acceso rápido a algunas aplicaciones de Office 2016, de modo que cualquier usuario que utilice el ordenador donde despluguemos la imagen personalizada, pueda acceder rápidamente a dichas aplicaciones.

En este punto se tiene una máquina virtual de referencia con el sistema operativo Windows 10 x64 Enterprise, la aplicación Office 2016 (ambos actualizados) y con el escritorio personalizado, preparada para su utilización.

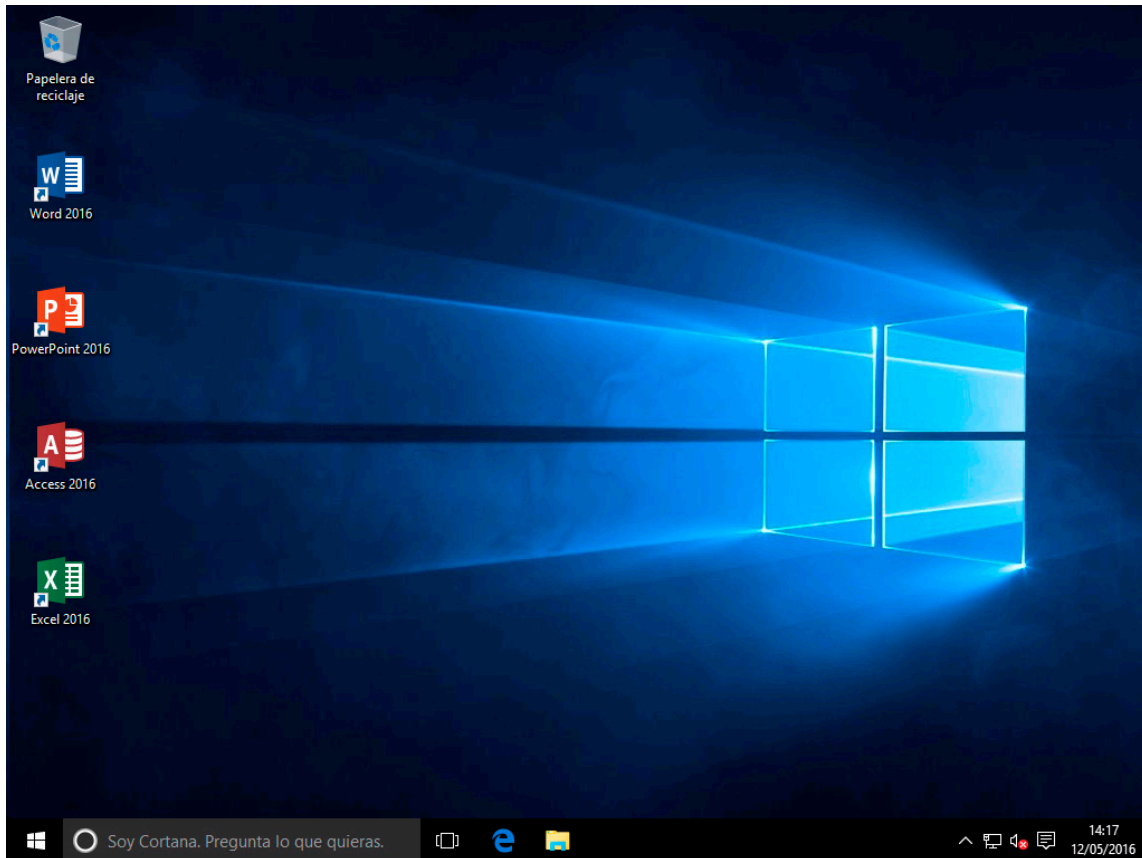


Figura 62: Escritorio personalizado para todos los usuarios en Windows 10.

4.6. Capturar la imagen de referencia personalizada

Para capturar la imagen de referencia se creó un DS. ¿Era necesario crear un DS específicamente para realizar la captura? La respuesta es no. Se pueden hacer capturas y despliegues en el mismo un DS, como el que se ha creado en el punto 4.4.1. (MDTLab), pero si mantenemos separados el DS de pruebas del DS de capturas, se pueden aplicar distintas configuraciones según se trate de un tipo de secuencia de despliegue o de captura.

4.6.1. Crear un DS para realizar capturas de imágenes

Se siguen los mismos pasos que los del punto 4.4.1., pero con los siguientes datos:

1. **Deployment share path:** c:\MDTCaptura
2. **Share name:** MDTCaptura\$
3. **Deployment share description:** MDT Captura
4. **Options:** se desmarcan
 - Ask if a computer backup should be performed
 - Ask if Bitlocker should be enabled

4.6.2. Importar un sistema operativo

Como el DS MDTCaptura solo se va a utilizar para realizar capturas de imágenes de referencia, MDT debe saber que versión de Windows va a capturar, para ello se debe de importar el mismo sistema operativo que tenía el ordenador de referencia. Los pasos a seguir son los mismos que en el punto 4.4.3., pero utilizando el DS MDTCaptura.

4.6.3. Crear una secuencia de captura de la imagen de referencia

Para crear una secuencia de tareas de captura de la imagen de referencia debemos seguir los pasos del punto 4.4.4., pero con la diferencia que ahora se selecciona la plantilla de secuencia **Sysprep and Capture** y se rellena con los siguientes datos:

1. **Task sequence ID:** REF01
2. **Task sequence name:** Captura imagen de referencia de Windows 10 x64 personalizada
3. **Task sequence comments:** Captura imagen de Windows 10 x64 Enterprise, con Office 2016, todo actualizado y con iconos de las aplicaciones de Office 2016 en el escritorio para todos los usuarios.
4. **Template:** Sysprep and Capture
5. **Select OS:** Windows 10 Enterprise (El que se ha importado en el paso 4.6.2.)
6. **Specify Product Key:** Do not specify a product key at this time
7. **Full Name:** ETSIAMN
8. **Organization:** UPV
9. **Internet Explorer Home Page:** www.upv.es
10. **Administrator password:** password

4.6.4. Ejecutar el fichero LiteTouch.vbs desde MDTCaptura

Importante: Se debe de crear un punto de control de la máquina virtual de Hyper-V antes de que lance la secuencia de captura. La secuencia de captura va a ejecutar el comando Sysprep [4.1.1.2.] y preparara la máquina para poder leer la imagen de referencia del ordenador. A un sistema operativo Windows se le pueden realizar un máximo de tres ejecuciones del comando Sysprep, después de las cuales, hay que restablecer la activación de Windows [21]. Si se hace de este modo se puede devolver la máquina virtual al punto de control creado anteriormente y realmente será como si no hubiésemos ejecutado nunca el comando Sysprep.

Desde la máquina de referencia se conectó al recurso de red MDTCaptura\$ del servidor MDT01. Para ello se solicitaron los siguientes datos:

- Usuario: MDT01\UsuarioMDT
- Contraseña: password

Para iniciar la captura se debe ejecutar el fichero LiteTouch.vbs, que está ubicado dentro de la carpeta Scripts del DS MDTCaptura. Se muestra un asistente para realizar la captura, en el que se selecciona lo siguiente:

- **Select a task sequence to execute on this computer:** Capturar imagen de referencia de Windows 10 x64 personalizada
- **Specify whether to capture an image:**
 - **Capture an image of this reference computer**
 - **Location:** \\MDT01\MDTCaptura\Captures , no lo modificamos
 - **File name:** REF01.wim
- **Specify credentials for connecting to network shares:**
 - **User Name:** administrador
 - **Password:** password
 - **Domain:** MDT01

Los datos que se pusieron para conectar al recurso compartido son del administrador del sistema. Esto fue necesario pues lo que hizo la secuencia de tareas es crear un fichero WIM dentro de la carpeta Captures del DS MDTCaptura, y para ello se necesita que el usuario tenga permisos de escritura en el punto de despliegue.

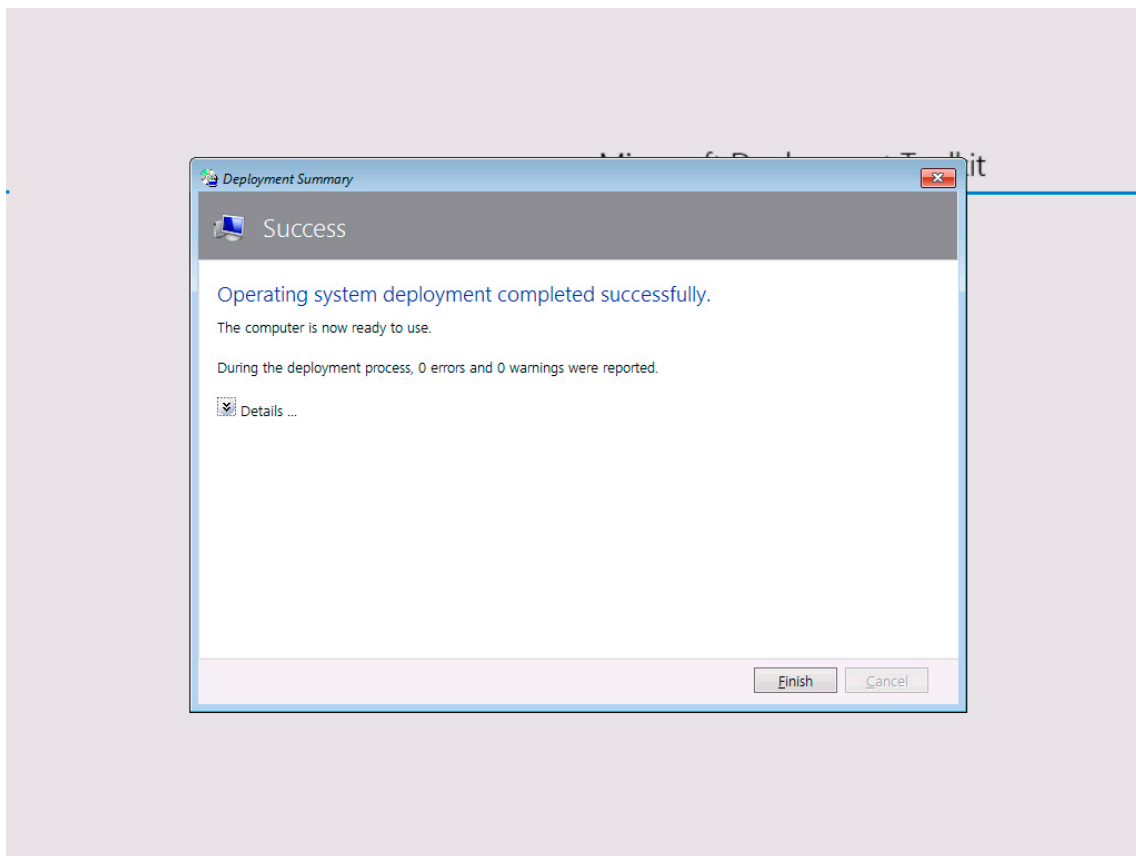


Figura 63: Mensaje de confirmación de la correcta captura de la imagen de referencia.

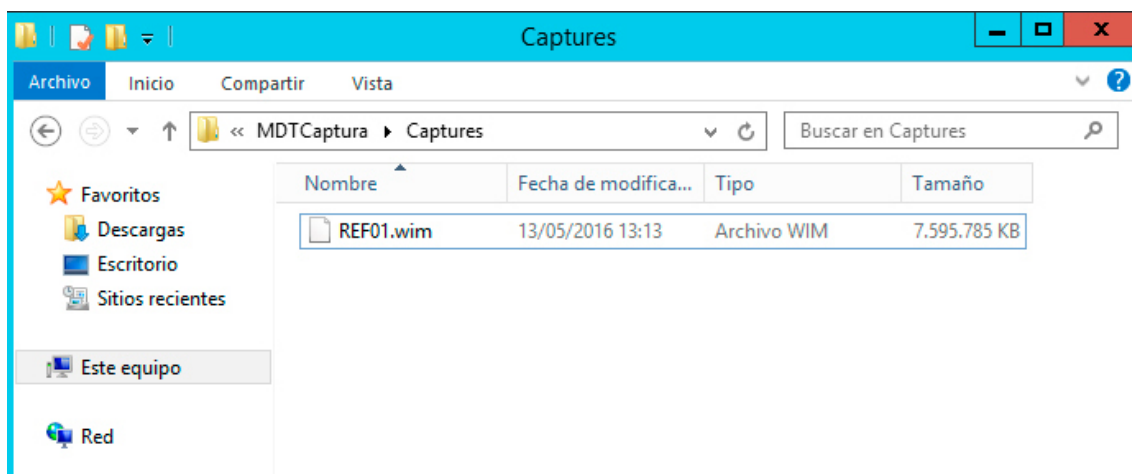


Figura 64: Explorador de archivos que muestra el fichero WIM capturado del ordenador de referencia.

La secuencia de captura de imagen ha ejecutado el comando Sysprep sobre la máquina de referencia. Para poder devolverla al estado de justo antes de ejecutar la secuencia de captura de la imagen, aplicamos el punto de control de la máquina de Hyper-V que se realizó anteriormente en el punto 4.6.4. Cuando la máquina termina de aplicar el punto de control, podemos apagarla.

4.7. Despliegue de la imagen de referencia

Para realizar el despliegue de la imagen de referencia utilizando el método Lite Touch, se importará la imagen de referencia al DS MDTLab y se creará una secuencia de tareas que nos permita desplegarla.

4.7.1. Importar la imagen de referencia en el DS MDTLab

Para poder importar la imagen de referencia, se siguieron los pasos del punto 4.4.3., pero se utilizaron los siguientes datos:

1. **Custom image file.** Se selecciona esta opción.
2. **Source file:** C:\MDTCaptura\Captures\REF01.WIM
3. **Move the files to the deployment share instead of copying them.** Se marca esta opción.
4. **Setup files are not needed.** Se selecciona esta opción.
5. **Destination directory name:** REF01

4.7.2. Crear secuencia de despliegue en el DS MDTLab

El proceso utilizado para crear la secuencia de despliegue, fue similar al descrito en el punto 4.4.4., pero el sistema operativo fue que se importó en el punto 4.7.1. (REF01.WIM):

1. **Task sequence ID:** DEPREF01
2. **Task sequence name:** Despliega Windows x64 personalizada
3. **Task sequence comments:** Despliega imagen personalizada REF01 con Windows 10 x64 Enterprise, Office 2016
4. **Select template:** Standard Client Task Sequence
5. **Select OS:** REF01.WIN (El que hemos importado en el paso 4.7.1.)
6. **Specify Product Key.** Seleccionamos la opción: Do not specify a product key at this time
7. **Full Name:** ETSIAMN
8. **Organization:** UPV
9. **Internet Explorer Home Page:** www.upv.es
10. **Administrator password:** password

4.7.3. Desplegar la imagen de referencia

Los pasos que se siguen para desplegar la imagen de referencia, son los mismos que en el punto 4.4.5., pero se utilizó la secuencia DEPREF01:

1. **Keyboard layout:** Spanish
2. **Run the Deployment Wizard to install a new Operating System**
3. **Deployment Share:** \\MDT01\MDTLab\$
4. **User name:** usuarioMDT
5. **Password:** password
6. **Domain:** MDT01
7. **Select task sequence to execute on this computer:** Despliega Windows x64 personalizada

4.8. Añadir aplicaciones y controladores en el DS MDTLab

Se pueden añadir aplicaciones que se instalarán bien utilizando la secuencia de tareas de despliegue, o bien configurando la base de datos de MDT, con una configuración específica para un ordenador. También se pueden agregar controladores de dispositivos al repositorio del DS, y que serán inyectados en el momento del despliegue, en función del hardware del ordenador cliente.

4.8.1. Añadir aplicaciones en el DS MDTLab

Se pueden añadir aplicaciones que se instalaran al final del despliegue. Una práctica que está totalmente desaconsejada en la creación de imágenes de referencia, es el integrar el software antivirus en la imagen. Como ejemplo, se agregará la aplicación antivirus Kaspersky al repositorio de aplicaciones del DS MDTLab.



Para poder instalar la aplicación sin ningún tipo de intervención, se necesita conocer la configuración de instalación silenciosa del software. En la página web de soporte del software antivirus Kaspersky, encontramos el parámetro que buscamos: **/s**, y un ejemplo de cómo iniciar una instalación silenciosa: "C:\kes 10.1\en\exec\setup.exe /s" [22].

Ahora ya se puede agregar el software de antivirus Kaspersky. Se seleccionó la carpeta Applications del DS MDTLab y con el botón derecho del ratón se seleccionó New Application. Siguiendo el asistente, se va completando con los siguientes datos:

1. Application Type

Se selecciona: Application with source files

2. Details

Application Name: Antivirus Kaspersky

3. Source

- Source directory: C:\Kasper
- No se selecciona la opción **Move the files to the deployment share instead of copying them**

4. Destination

Specify the name of the directory that should be created: Antivirus Kaspersky

5. Command Details

- Command line: `setup.exe /s`
- Working directory: `.\Applications\Antivirus Kaspersky`

Ahora se debe modificar la secuencia de tareas para que se instale el antivirus a todos los ordenadores donde se ejecute el despliegue. Para ello se selecciona la siguiente secuencia dentro de DS MDTLab:

- Nombre de la secuencia: Despliega Windows x64 personalizada
- ID: DEPREF01

Cuando visualizamos la secuencia de tareas vemos que está organizada por carpetas que son las diversas fases de la secuencia. Cada carpeta o fase, contiene los diversos pasos o tareas que se deben de ejecutar, e incluso puede contener otras carpetas o fases. El antivirus debe de ser lo último que se instale en la fase final del despliegue. La última fase se llama State Restore, y dentro esta carpeta esta la fase Custom Tasks. Es aquí donde se crea un nuevo paso para instalar el antivirus automáticamente. Se seleccionó en el menú superior de las propiedades de la tarea lo siguiente: Add, General, Install Application, y se rellenó con los siguientes datos:

- 1. Name:** Instalar Kaspersky
- 2. Install a single application.** Se seleccionó
- 3. Application to install,** Se pulsó en el botón Browse
4. Se seleccionó la aplicación Antivirus Kaspersky

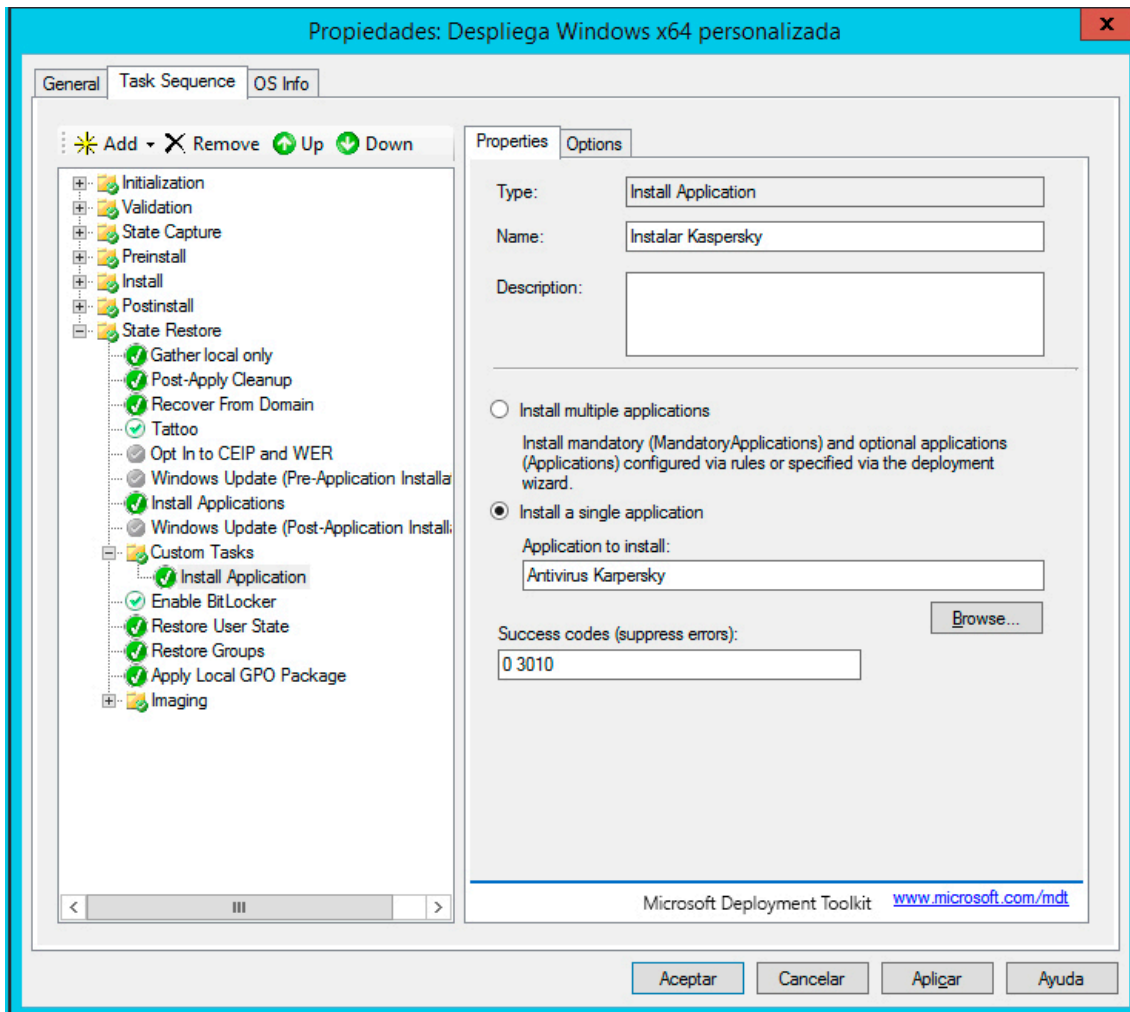


Figura 65: Selección de la aplicación antivirus Kaspersky.

Se realizó un despliegue LTI de pruebas y se comprueba que el antivirus Kaspersky se ha instalado automáticamente al final del despliegue.

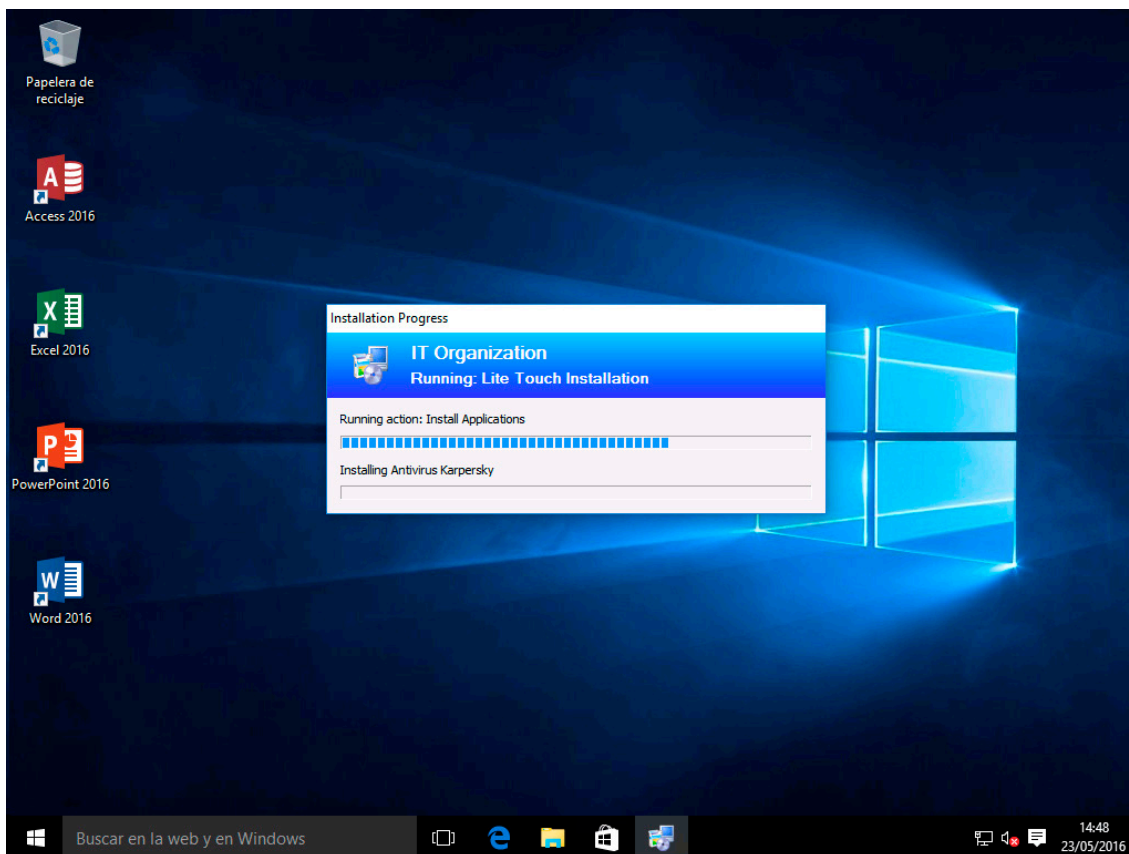


Figura 66: Instalación desatendida del antivirus Kaspersky al final del despliegue.

4.8.2. Añadir controladores de dispositivos en el DS MDTLab

Para simplificar las cosas se recomienda elegir uno de los tres escenarios posibles para configurar los controladores de dispositivos en MDT. Los tres escenarios se basan en el tamaño de la empresa, el número de sistemas operativos que tienen que ser desplegados, el nivel de control deseado, y el número de los distintos modelos de hardware [23].

Escenario número 1: caos total

Este escenario tiene las siguientes suposiciones. Se trata de una empresa pequeña, que sólo implementa un sistema operativo, por ejemplo, Windows 8.1 x64, y tienen unos pocos modelos de hardware del mismo fabricante. La clave aquí es que se está desplegando sólo una familia de sistemas operativos y que el hardware es del mismo fabricante. La razón es que los grandes fabricantes de hardware, hacen más pruebas de compatibilidad entre sus propios modelos por cada familia de sistema operativo, por lo que es bastante raro que un controlador de un modelo de hardware interfiera con otro.

Solución del escenario número 1

Para este escenario se recomienda utilizar el método basado en la detección de PnP ID. Lo único que hay que hacer es descargar y extraer los controladores para cada modelo en una carpeta, e importar esa carpeta en el Deployment Workbench.

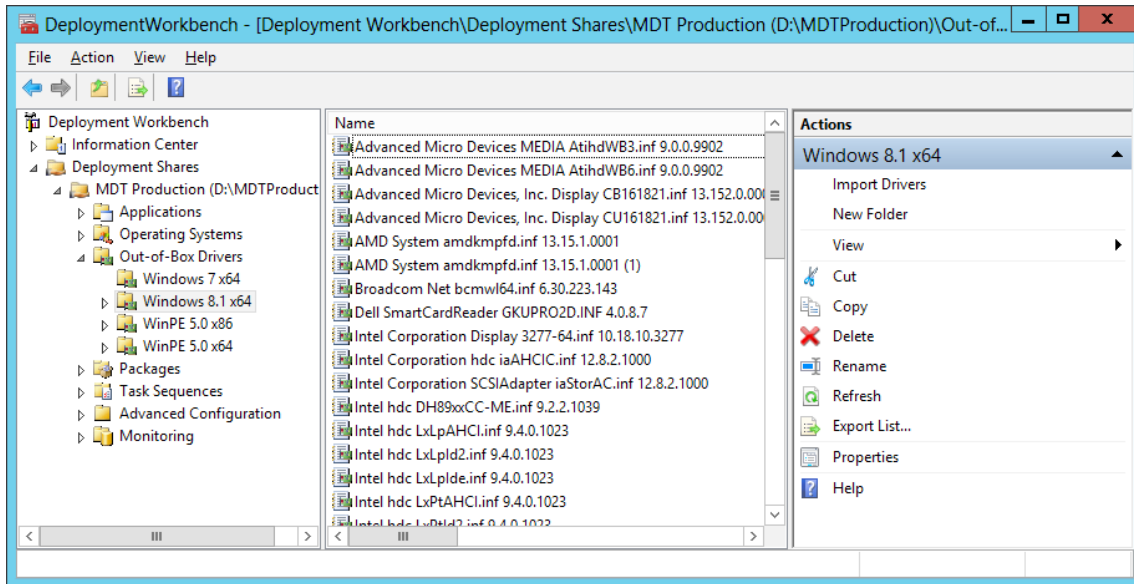


Figura 67: Los controladores importados en el Deployment Workbench.

Escenario número 2: añadiendo predictibilidad

Este escenario parte de las siguientes suposiciones: se trata de una empresa pequeña o mediana, que están desplegando múltiples sistemas operativos, por ejemplo, Windows 7 x64 SP1 y Windows 8.1 x 64, y tienen unos cuantos modelos distintos de hardware pero del mismo fabricante. La principal diferencia con el primer escenario es que están desplegando múltiples sistemas operativos. Dado que el método por defecto que se utiliza MDT es la detección de los identificadores PnP entre todos los controladores de dispositivos importados, necesitamos tener una forma de filtrar los controladores de manera que sólo los controladores de Windows 7 sean considerados para implementaciones de Windows 7 SP1, y que sólo los controladores de Windows 8.1 se utilicen con Windows 8.1. La característica de MDT que podemos utilizar para este filtrado se denomina *Profile Selection*, o perfil de selección.

Solución del escenario número 2

Para este escenario se recomienda utilizar el método basado en la detección predeterminada de los identificadores PnP, con el añadido de usar los perfiles de selección como un filtro para los controladores de dispositivos. Lo que tenemos que hacer es crear una carpeta para cada sistema operativo dentro de Out-of-box Drivers, una denominada Windows 7 x64, y otra Windows 8.1 x64. Después tenemos que importar los controladores de dispositivos correctos para cada sistema operativo dentro de su respectiva carpeta.

Después de importar los controladores, hay que crear un perfil de selección para cada una de las carpetas de controladores de cada sistema operativo y, a continuación, hay que configurar la acción *Inject Controllers*, o inyección de controladores en la secuencia de tareas para que utilice el perfil de selección correcto.

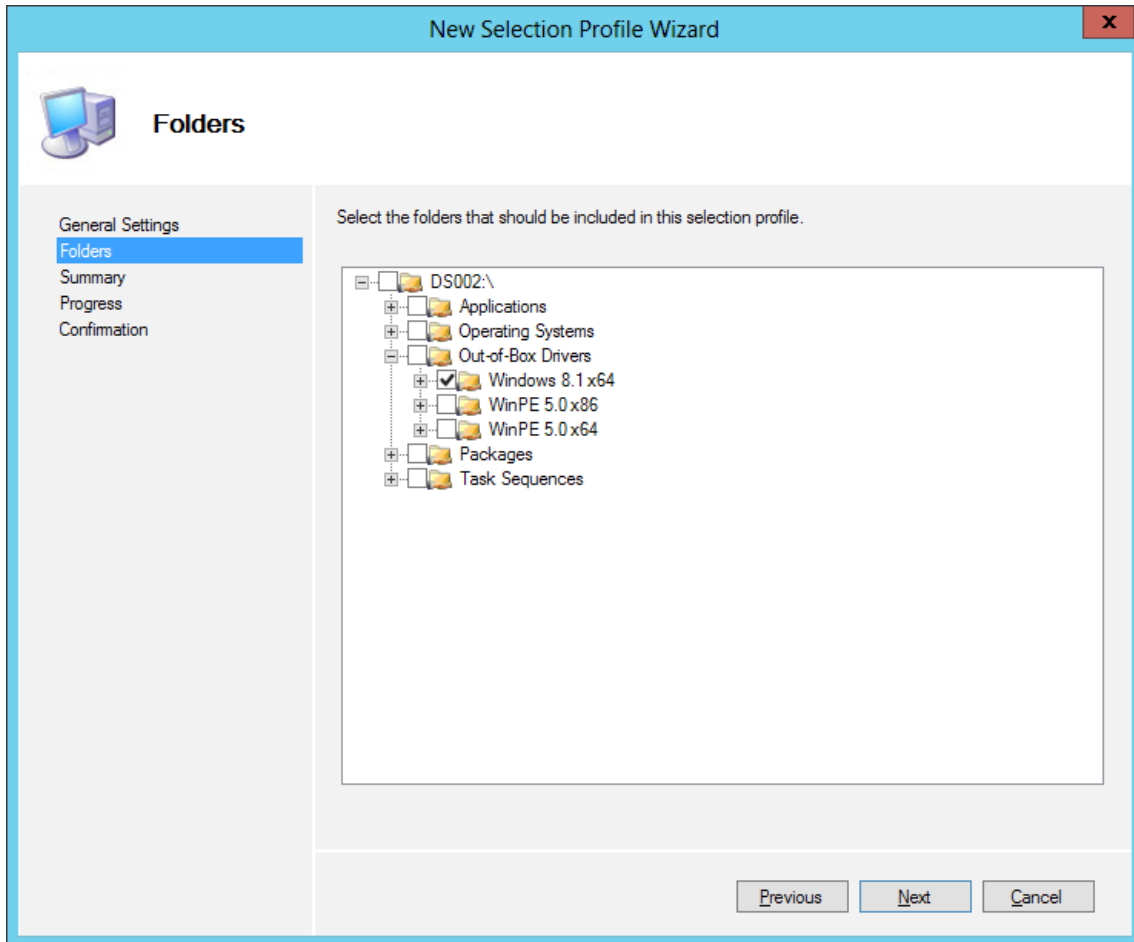


Figura 68: Creación del perfil de selección de Windows 8.1 x64.

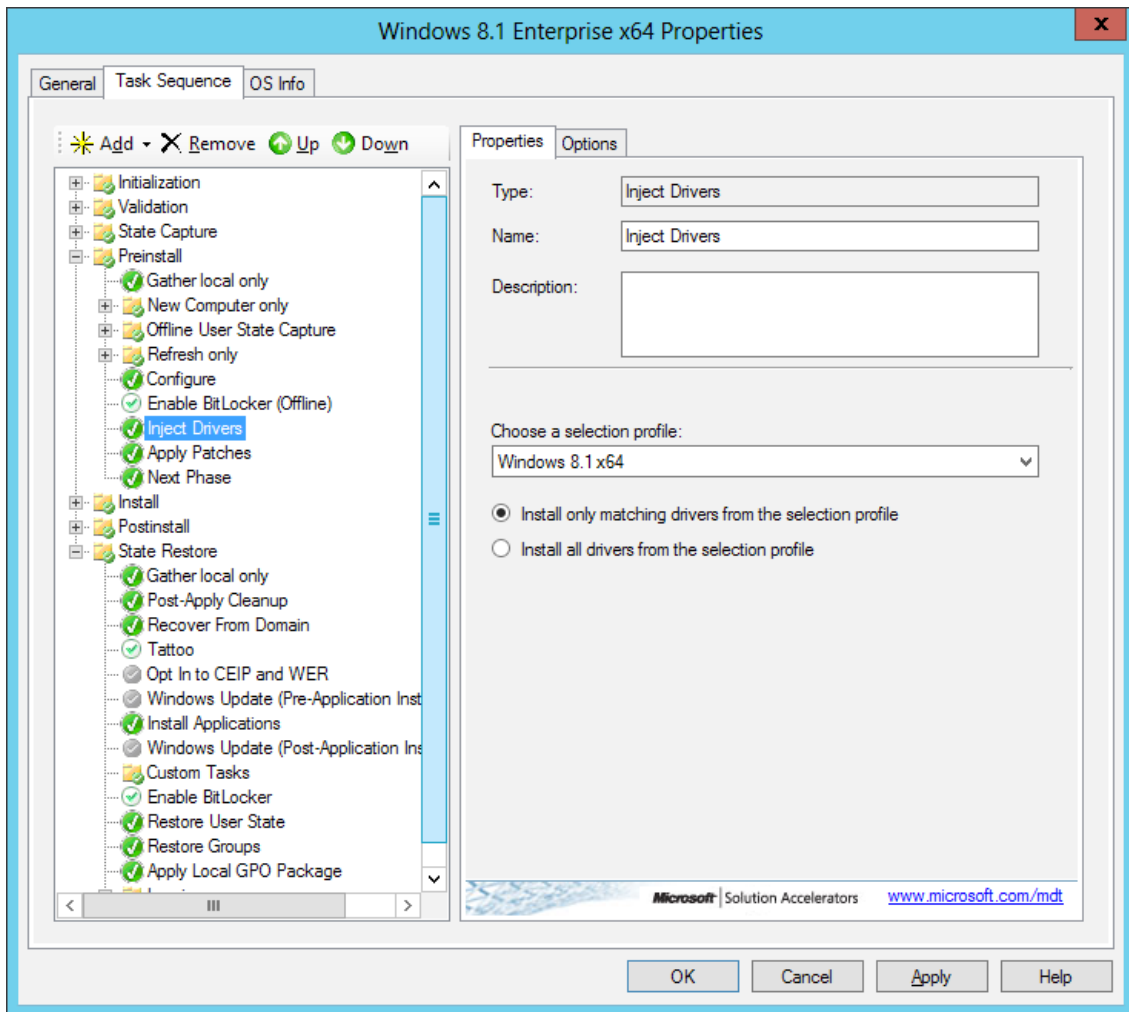


Figura 69: Configuración de la acción de inyectar controladores para que utilice el perfil de selección de Windows 8.1 x64.

Escenario número 3: control total

En este escenario partimos de las siguientes suposiciones: se trata de una empresa pequeña, mediana o grande, que están desplegando múltiples sistemas operativos, por ejemplo, Windows 7 SP1 y Windows 8.1, tienen muchos modelos de hardware y de múltiples fabricantes. La principal diferencia con respecto al segundo escenario, es que están utilizando hardware de múltiples fabricantes, y el hecho de que quieren un control más detallado de sus conductores.

Nota: Johan Arwidmark (autor del artículo) opina que este método es su favorito, porque, aunque se necesita más tiempo para configurarlo, se consigue un control completo de todo el proceso de inyección de controladores de dispositivos [23].

Como hay múltiples fabricantes involucrados, la compatibilidad entre diferentes modelos de hardware, no se puede garantizar únicamente basándose en la detección de identificadores de PnP. Hay que ser capaz de filtrar no sólo en el sistema operativo, sino

también en función de cada modelo de hardware. A pesar de que técnicamente se podría utilizar perfiles de selección para esto también, no es para lo que fueron diseñados. Hay otra característica llamada *DriverGroup* que ayudará a hacer un filtrado más avanzado.

Solución del escenario número 3

Para este escenario se recomienda no utilizar el método basado en la detección por defecto de los identificadores PnP, y en su lugar utilizar *DriverGroup* como filtro para los controladores. Lo primero es crear dos carpetas dentro de la carpeta Out-of-box drives, por ejemplo, una con el nombre de Windows 7 x64 y otra con el nombre Windows 8.1 x64. A continuación, dentro de cada carpeta de los sistemas operativos, creamos carpetas para cada modelo de hardware que tengamos. Después tenemos que descargar y extraer los controladores para cada modelo, y para cada sistema operativo. Luego importamos los controladores en la carpeta que corresponda, teniendo en cuenta la estructura de sistema operativo y modelo de hardware.

Hay que tener en cuenta que los perfiles de selección y *DriverGroup* trabajan conjuntamente, es decir, si hay un perfil de selección que incluye el controlador A, y un *DriverGroup* que tiene incluido el controlador B, se añadirán los dos controladores. La mayoría de las veces sólo necesitamos un controlador u otro, pero si nos interesa se pueden combinar. Al utilizar este método, es recomendable que se utilice el perfil de selección **Nothing** en la acción **Inject Drivers**. El verdadero truco para este escenario es nombrar las carpetas del controlador de acuerdo con el nombre del modelo de hardware, porque entonces se puede establecer una variable de entorno *DriverGroup001*, como %Model%, bien en la secuencia de tareas o en las reglas.

Además, para no tener problemas con los controladores que no han sido detectados por PnP dentro de la carpeta (*DriverGroup*), se recomienda forzar la inyección de controladores cambiando la propiedad de la acción **Inject Drivers** a **Install all drivers from the selection profile**.

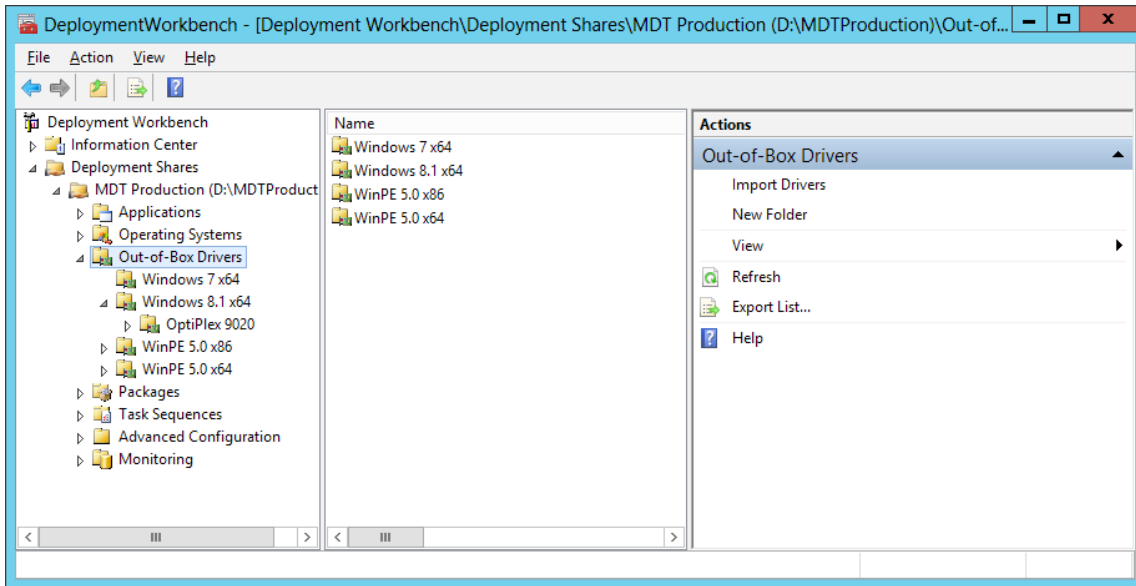


Figura 70: Creación de la estructura de las carpetas de controladores en el Deployment Workbench.

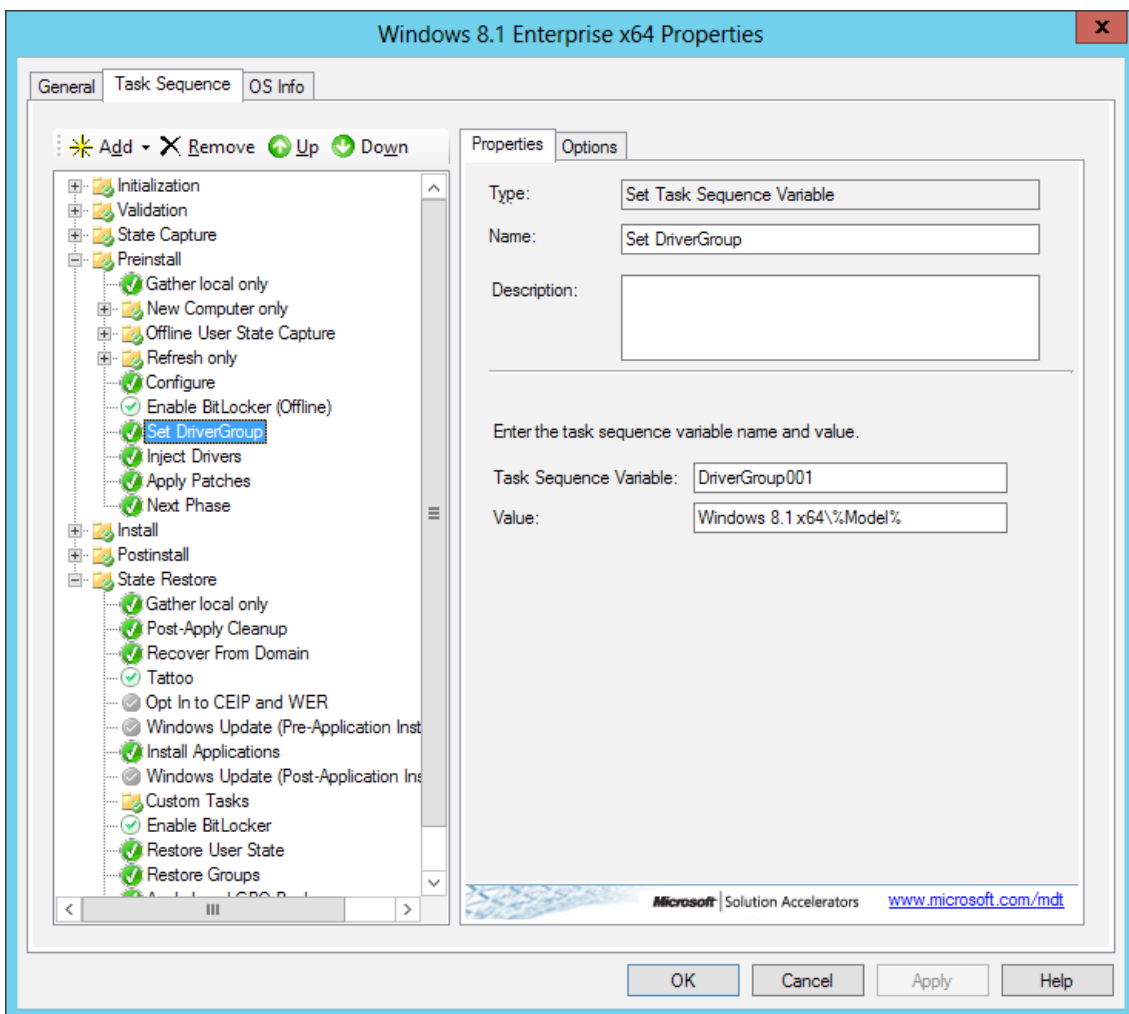


Figura 71: Añadir la acción de establecer la variable DriverGroup001 de la secuencia de tareas con el valor: Windows 8.1 x 64 \ % Model %.



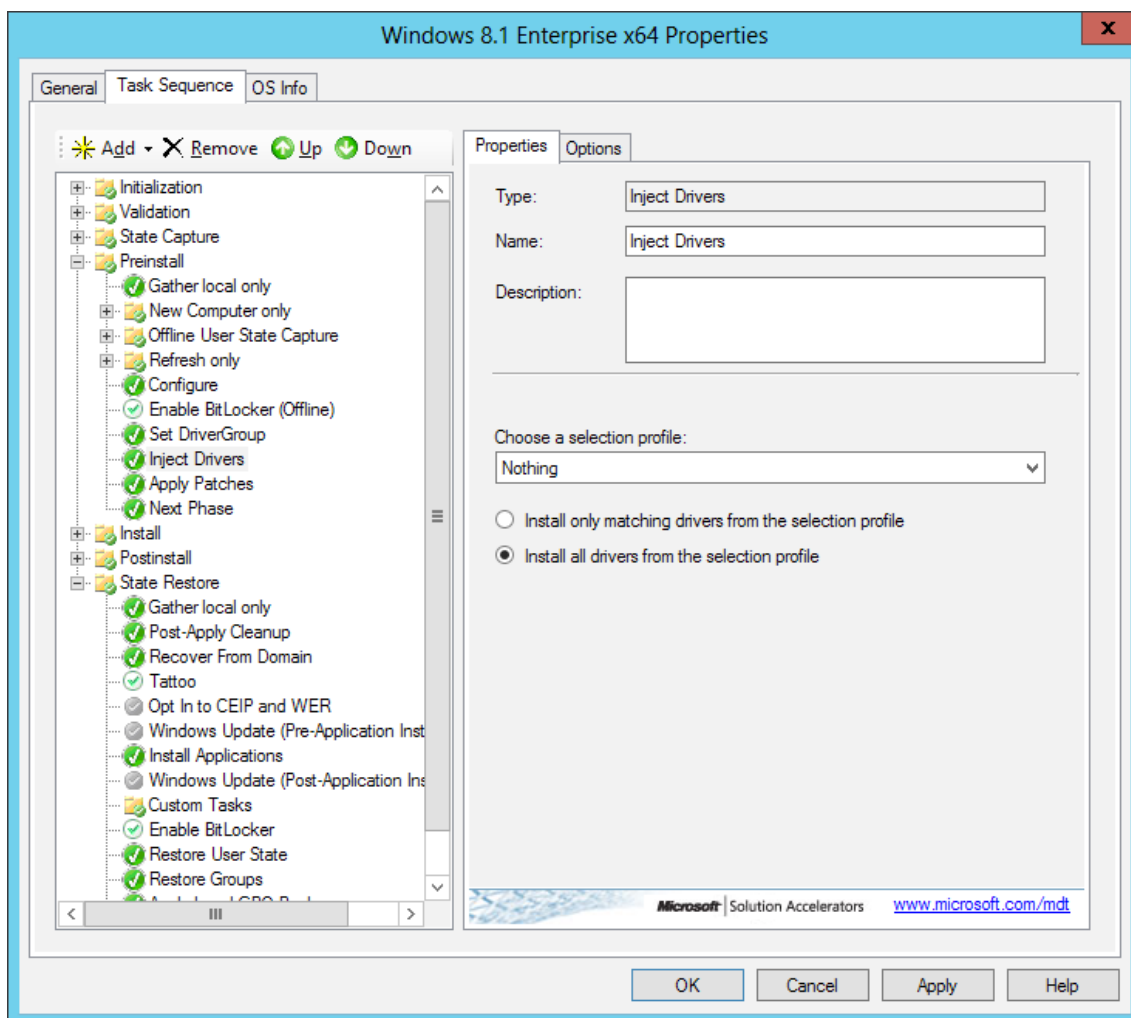


Figura 72: Configurar la acción de inyectar controladores a el perfil de selección *Nothing*.

4.9. Automatizando el despliegue. Pasar de utilizar LTI a utilizar ZTI

Los despliegues LTI requieren que el usuario o el técnico que está realizando el despliegue interactúe con la aplicación, aportando los datos necesarios por medio de un asistente. Para conseguir hacer despliegues ZTI con MDT, se tiene que añadir información adicional en la configuración del punto de despliegue y en una base de datos de SQL que utilizara MDT para pre-almacenar los datos específicos de cada ordenador.

4.9.1. Rules (CustomSettings.ini) y BootStrap.ini

En este punto del proceso se debe cambiar la configuración del DS MDTLab, seleccionando las propiedades del mismo. Dentro de la pestaña Rules encontramos un editor de texto, que muestra la configuración actual. Realmente este editor esta visualizando el fichero CustomSettings.ini, que está en la carpeta Control dentro del DS MDTLab.

```

1 [Settings]
2 Priority=Default
3 Properties=MyCustomProperty
4
5 [Default]
6 ; Instalar un sistema operativo = SI
7 OSInstall=Y
8 ; No preguntar si capturar una imagen = NO
9 SkipCapture=NO
10 ; No preguntar el password del Administrador = SI
11 SkipAdminPassword=YES
12 ; No preguntar la clave de producto = SI
13 SkipProductKey=YES
14 ; No preguntar el hacer un copia de seguridad del ordenador = NO
15 SkipComputerBackup=NO
16 ; No preguntar si activar Bitlocker = SI
17 SkipBitLocker=NO

```

Código 1: Contenido del fichero CUSTOMSETTINGS.INI del DS MDTLab.

Se añadieron más líneas de configuración al fichero para conseguir hacer despliegues ZTI, quedando de la siguiente manera:

```

1 [Settings]
2 Priority=Default
3 Properties=MyCustomProperty
4
5 [Default]
6 ; Instalar un sistema operativo = SI
7 OSInstall=Y
8 ; No preguntar el password del Administrador = SI
9 SkipAdminPassword=YES
10 ; No preguntar la clave de producto = SI
11 SkipProductKey=YES
12 ; No preguntar el hacer un copia de seguridad del ordenador = SI
13 SkipComputerBackup=YES
14 ; No preguntar si activar Bitlocker = SI
15 SkipBitLocker=YES
16 ; No preguntar el nombre del ordenador =SI
17 SkipComputerName=YES
18 ; No preguntar por agregar a un dominio = SI
19 SkipDomainMembership=YES
20 ; No preguntar por guardar y recuperar los datos del usuario = SI
21 SkipUserData=YES
22 ; Ubicacion para los ficheros de datos del usuario = NINGUNA
23 UserDataLocation=NONE
24 ; No preguntar si captura una imagen = SI
25 SkipCapture=YES
26 ; Realizar captura de imagen = NO
27 DoCapture=NO
28 ; No preguntar por la configuración regional = SI
29 SkipLocaleSelection=YES
30 ; No preguntar que secuencia desplegar = SI
31 SkipTaskSequence=YES
32 ; No preguntar por la zona horaria = SI
33 SkipTimeZone=YES

```



```
34 ; No preguntar que aplicaciones instalar = SI
35 SkipApplications=YES
36 ; No mostrar el resumen de información del despliegue = SI
37 SkipSummary=YES
38 ; No mostrar la pantalla de bienvenida = SI
39 SkipBDDWelcome=YES
40 ; Identificador de la tarea a desplegar = REF01
41 TaskSequenceID=W10AULAX64
42 ; Especificamos el código y el nombre de la zona horaria del ordenador
43 ; España es 001 y Romance Standard Time
44 TimeZone=001
45 TimeZoneName=Romance Standard Time
46 ; Especificamos el idioma del teclado, del ordenador, del interface
grafico
47 KeyboardLocale=es-ES
48 UserLocale=es-ES
49 ImageLanguage001=es-ES
50 UILanguage=es-ES
51 ; Especificamos el dominio donde se unirá el ordenador
52 JoinDomain=alumno
53 ; Especificamos el usuario con privilegios para agregar ordenadores al
dominio
54 DomainAdmin=agregar
55 DomainAdminDomain=alumno
56 DomainAdminPassword=*****
57 ; Especificamos los usuarios que serán administradores del ordenador
58 Administrators001=alumno\adminMDT
59 ; Especificamos la ubicación donde guardar los logs del despliegue en el
servidor
60 SLShare=%DeployRoot%\Captures\Log
61 ; No mostrar el resumen final del despliegue = SI
62 SkipFinalSummary=YES
63 ; Accion a ejecutar al final del despliegue = REINICIAR
64 FinishAction=REBOOT
65 ; Servicio de monitorización del despliegue
66 EventService=http://MDT01:9800
```

Código 2: Contenido del fichero CUSTOMSETTINGS.INI para hacer ZTI.

Nota: la línea 62 hace que el ordenador no muestre el resumen final con el resultado del despliegue, y la línea 64 reinicia el ordenador cuando termina el despliegue. Estas dos líneas se deberían añadir cuando terminemos de realizar todas las pruebas de despliegue.

Ahora el fichero BootStrap.ini:

```
1 [Settings]
2 ; Prioridades a cumplir
3 ; En modo LTI solo esta la de por defecto = Default
4 Priority=Default
5
6 [Default]
7 ; Ubicación en la red del punto de despliegue
8 ; en formato UNC = \\servidor\recursocompartido
9 DeployRoot=\\MDT01\MDTlab$
```

Código 3: Contenido del fichero BOOTSTRAP.INI del DS MDTLab.

Con las modificaciones para hacer ZTI.

```

1 [Settings]
2 ; Prioridades a cumplir
3 ; En modo LTI solo esta la de por defecto = Default
4 Priority=Default
5
6 [Default]
7 ; Ubicación en la red del punto de despliegue
8 ; en formato UNC = \\servidor\recursocompartido
9 DeployRoot=\\MDT01\MDTLab$
10 ; Usuario con permisos de acceso y lectura al punto de despliegue
11 UserID=UsuarioMDT
12 ; Dominio del usuario
13 UserDomain=MDT01
14 ; Password del usuario
15 UserPassword=password
16 ; Especificamos el idioma local del teclado
17 KeyboardLocale=es-ES
18 ; No mostrar la pantalla de bienvenida
19 SkipBDDWelcome=YES

```

Código 4: Contenido del fichero BOOTSTRAP.INI para hacer ZTI.

4.9.2. WMI y Windows Powershell

Cuando comienza la ejecución de una secuencia de tareas, uno de los primeros cometidos que realiza es la recopilación de información del ordenador donde se ejecuta. El *script* que se encarga de ello es ZTIGather.wsf. La información que recopila se refiere al hardware, como, por ejemplo:

- La marca del ordenador.
- El modelo del ordenador.
- Su UUID.
- La cantidad de memoria.
- La arquitectura del procesador.

Toda esa información la reúne interrogando al Windows Management Instrumentation (WMI) del ordenador utilizando comandos de Windows Powershell.

4.9.2.1. ¿Qué es WMI?

Es una tecnología de gestión de sistema de Windows; se puede utilizar WMI para administrar equipos locales y remotos. WMI proporciona un enfoque consistente para llevar a cabo las tareas de gestión del día a día con los lenguajes de programación o de *scripting*. Por ejemplo, se puede: [24]

- Iniciar un proceso en un equipo remoto.
- Programar un proceso que se ejecute en momentos específicos, en días específicos.

- Reiniciar un ordenador de forma remota.
- Obtener una lista de las aplicaciones instaladas en un equipo local o remoto.
- Consultar los registros de eventos de Windows en un equipo local o remoto.
- Recopilar información acerca del hardware del ordenador.

4.9.2.2. ¿Qué es Windows Powershell?

Windows PowerShell es un lenguaje de *scripting* y *shell* de línea de comandos basado en tareas diseñado especialmente para la administración del sistema. Creado con Microsoft .NET Framework, Windows PowerShell ayuda a los profesionales de TI y a los usuarios experimentados a controlar y automatizar la administración tanto del sistema operativo Windows como de las aplicaciones que se ejecutan en Windows.

Los comandos de Windows PowerShell, denominados *cmdlets*, permiten administrar los equipos de la empresa desde la línea de comandos. Los proveedores de Windows PowerShell permiten obtener acceso a almacenes de datos (como el Registro y el almacén de certificados) con la misma simplicidad con que se obtiene acceso al sistema de archivos. Además, Windows PowerShell tiene un analizador de expresiones muy completo y un lenguaje de *scripting* totalmente desarrollado [25].

4.9.2.3. Utilizar Windows Powershell para obtener información del ordenador

Como ya hemos dicho, cuando se realiza un despliegue de una imagen, se lanza un *script* que recopila el UUID, la dirección MAC de la tarjeta de red (*MAC Address*) y el número de serie del ordenador (*Serial Number*). Estos tres datos son únicos para cada ordenador. MDT utilizará esos datos para buscar en la base de datos alguna coincidencia, y de encontrar alguna, utilizará la información del ordenador almacenada en la base de datos en el momento del despliegue. De este modo, si la base de datos devuelve:

- Cero ocurrencias: el ordenador es desconocido en la base de datos de MDT.
- Una ocurrencia: la base de datos de MDT tiene los datos del ordenador y es conocido.
- Más de una ocurrencia: tenemos el mismo registro repetido en la base de datos.

Si se produjera un error por esta causa durante el despliegue, esta información la se puede consultar posteriormente en los ficheros de registro del ordenador. El resultado de la búsqueda del ordenador en la base de datos, junto con el resto de información del despliegue, queda grabado en unos ficheros de registro. Cada ordenador tiene su propio grupo de ficheros de registro, que se generan localmente en el ordenador durante el despliegue, y si las reglas del DS están correctamente configuradas, se copian en el servidor. De este modo si se produce un error a la hora de encontrar el ordenador en la base de datos de MDT, se puede revisar dichos archivos de registro en busca de una posible solución.

La línea 62, del Código 2, `SLShare=%DeployRoot%\Captures\Logs`, indica a la secuencia de tareas que se guarde los ficheros de registro dentro del recurso compartido del punto de despliegue, donde se ha ejecutado la secuencia, en la ruta `Captures\Logs`.

Para agregar un ordenador nuevo a la base de datos de MDT, se necesita conocer de antemano al menos uno de los tres datos del ordenador:

- Su UUID.
- Su dirección MAC.
- Su número de serie.

Para averiguar estos datos vamos a utilizar comandos de Windows Powershell.

4.9.2.3.1. Obtener el UUID y el número de serie del ordenador

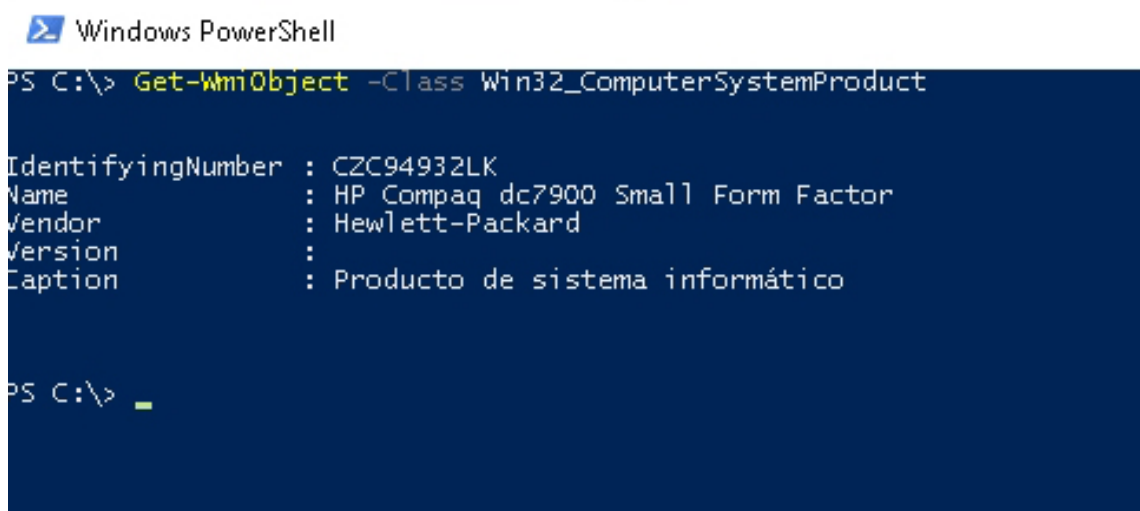
Para averiguar el UUID del ordenador haremos uso de la clase `Win32_ComputerSystemProduct` que representa un producto. Esto incluye el software y el hardware del ordenador. Esta clase tiene las siguientes propiedades [26]:

- *Caption*: es una descripción breve del producto.
- *Description*: descripción del producto.
- *IdentifyingNumber*: número identificativo, como pueda ser un número de serie de un software o un número de serie grabado en el hardware.
- *Name*: nombre común del producto.
- *SKUNumber*: número SKU del producto.
- *UUID*: identificador único universal para este producto. Es un identificador de 128 bits que se garantiza que sea diferente de otros UUID generados. Si un UUID no está disponible, se utiliza un UUID de todos ceros.
- *Vendor*: nombre del proveedor del producto, puede ser el nombre del fabricante, revendedor, o OEM.
- *Version*: información de la versión del producto.

Si en una ventana de Windows Powershell se ejecuta el siguiente comando:

```
Get-WmiObject -Class Win32_ComputerSystemProduct
```

y se obtiene el resultado de la Figura 73.



```
Windows PowerShell
PS C:\> Get-WmiObject -Class Win32_ComputerSystemProduct

IdentifyingNumber : CZC94932LK
Name              : HP Compaq dc7900 Small Form Factor
Vendor            : Hewlett-Packard
Version           :
Caption           : Producto de sistema informático

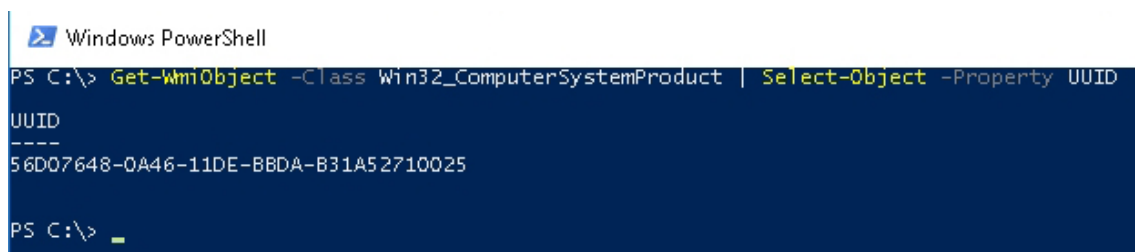
PS C:\> _
```

Figura 73: Ejecución del comando `Get-WmiObject -Class Win32_ComputerSystemProduct`.

Como vemos en la Figura 73, hemos obtenido cinco de las nueve propiedades de la clase. Ya tenemos el número de serie (*IdentifyingNumber*), pero no el UUID. Para recuperar el UUID tenemos que solicitar que nos devuelva explícitamente esa propiedad de la clase, para ello se ejecutó el siguiente comando:

```
Get-WmiObject -Class Win32_ComputerSystemProduct | Select-Object -Property UUID
```

Y se obtuvo el UUID del ordenador (ver Figura 74).



```
Windows PowerShell
PS C:\> Get-WmiObject -Class Win32_ComputerSystemProduct | Select-Object -Property UUID

UUID
----
56D07648-0A46-11DE-BBDA-B31A52710025

PS C:\> _
```

Figura 74: Ejecución del comando `Get-WmiObject -Class Win32_ComputerSystemProduct | Select-Object -Property UUID`.

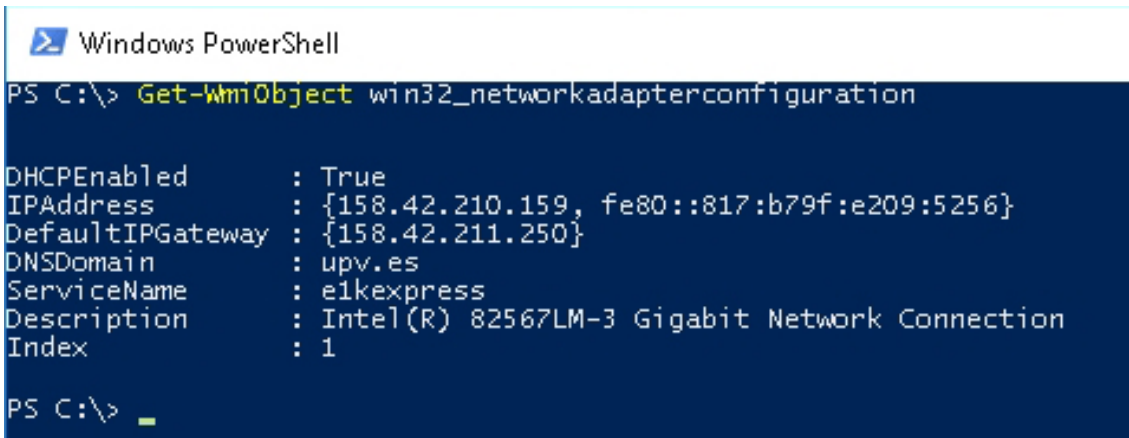
4.9.2.3.2. Obtener la dirección MAC del adaptador de red

La clase WMI `Win32_NetworkAdapterConfiguration` representa los atributos y características de un adaptador de red. Esta categoría comprende las propiedades adicionales y métodos de soporte de la gestión del protocolo TCP/IP que son independientes del adaptador de red [27].

La propiedad que nos interesa averiguar es `MACAddress`: Media Access Control (MAC), es la dirección física del adaptador de red. La dirección MAC es asignada por el fabricante de modo que identifica unívocamente al adaptador de red.

Para obtener esa información se ejecuta la siguiente orden en Windows Powershell:

```
Get-WmiObject Win32_NetworkAdapterConfiguration
```



```
Windows PowerShell
PS C:\> Get-WmiObject win32_networkadapterconfiguration

DHCPEnabled      : True
IPAddress        : {158.42.210.159, fe80::817:b79f:e209:5256}
DefaultIPGateway : {158.42.211.250}
DNSDomain        : upv.es
ServiceName      : e1kexpress
Description      : Intel(R) 82567LM-3 Gigabit Network Connection
Index            : 1

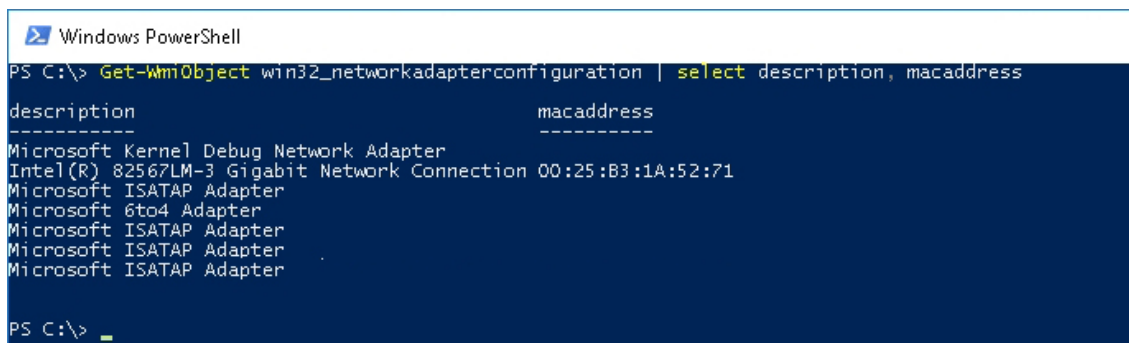
PS C:\> _
```

Figura 75: Ejecución del comando `Get-WmiObject Win32_NetworkAdapterConfiguration`.

Como se observa en la Figura 75 se muestran siete propiedades, pero no obtuvimos la dirección MAC. Se debe de solicitar específicamente la propiedad `MACAddress` con el siguiente comando:

```
Get-WmiObject Win32_NetworkAdapterConfiguration | Select Description,Macaddress
```

Una vez se ejecuta el comando anterior, se obtiene la dirección MAC del adaptador de red, junto con su descripción (ver Figura 76).



```
Windows PowerShell
PS C:\> Get-WmiObject win32_networkadapterconfiguration | select description, macaddress

description                               macaddress
-----
Microsoft Kernel Debug Network Adapter
Intel(R) 82567LM-3 Gigabit Network Connection 00:25:B3:1A:52:71
Microsoft ISATAP Adapter
Microsoft 6to4 Adapter
Microsoft ISATAP Adapter
Microsoft ISATAP Adapter
Microsoft ISATAP Adapter

PS C:\> _
```

Figura 76: Ejecución del comando `Get-WmiObject Win32_NetworkAdapterConfiguration | Select Description, Macaddress`.

4.9.2.3.3. Obtener más información útil sobre el ordenador

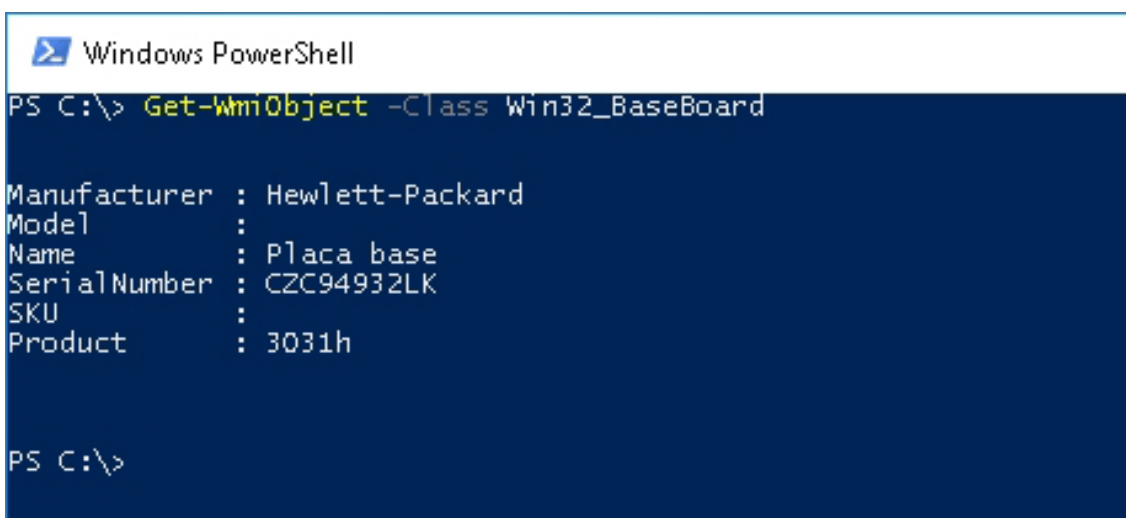
En algunos casos la información que los fabricantes incorporan en el WMI suele ser incompleta o insuficiente. Es importante que cuando MDT realiza el despliegue de una imagen, se determine correctamente de que modelo de ordenador se trata, de este modo se inyectarán los controladores de dispositivos adecuados. Los datos que identifican al ordenador en el ejemplo que hemos utilizado, están completos, pero

podemos encontrarnos con que no es suficiente con la información que devuelve la clase Win32_ComputerSystemProduct [28].

Para conseguir más información, se puede utilizar la clase WMI Win32_BaseBoard, que tiene veintinueve propiedades. Las propiedades que nos interesan son el fabricante y el modelo del ordenador. Si se ejecuta el siguiente comando de Windows Powershell

```
Get-WmiObject -Class Win32_BaseBoard
```

obtenemos el siguiente resultado (ver Figura 77).



```
Windows PowerShell
PS C:\> Get-WmiObject -Class Win32_BaseBoard

Manufacturer : Hewlett-Packard
Model       :
Name        : Placa base
SerialNumber : CZC94932LK
SKU         :
Product     : 3031h

PS C:\>
```

Figura 77: Ejecución del comando `Get-WmiObject -Class Win32_BaseBoard`.

Se nos muestran los siguientes datos:

- *Manufacturer*: nombre de la organización responsable de la fabricación del elemento físico.
- *Model*: nombre por el que es conocido el elemento físico.
- *Name*: etiqueta por la que se conoce el objeto.
- *SerialNumber*: número de serie utilizado para identificar el elemento físico.
- *SKU*: número SKU del elemento físico.
- *Product*: número de la placa base definido por el fabricante.

4.9.3. Base de datos de MDT en SQL Server

Con toda la información recopilada sobre el ordenador, ya se pueden añadir ordenadores a la base de datos MDT de SQL Server. Para ello hay que crear una nueva base de datos dentro de la instancia de SQL 2014 Express para que MDT pueda almacenar los datos de los ordenadores en los que posteriormente desplegaremos las imágenes.

4.9.3.1. Crear la base de datos

Para crear la base de datos dentro del DS que queremos utilizar para hacer los despliegues (MDTLab) hay que expandir la carpeta Advanced Configuration, y sobre la carpeta Database pulsar con el botón derecho del ratón y New Database.

Se pusieron los siguientes datos:

1. **SQL Server name:** MDT01
2. **Instance:** SQLExpress
3. **Network Library:** Named Pipes
4. **Create a new database** y le damos el nombre MDT
5. **SQL Share:** MDTLab\$

La base de datos creada en el DW nos permite almacenar información de ordenadores (Computers), roles (Roles), localizaciones (Locations) y marcas y modelos de ordenadores (Make and Model).

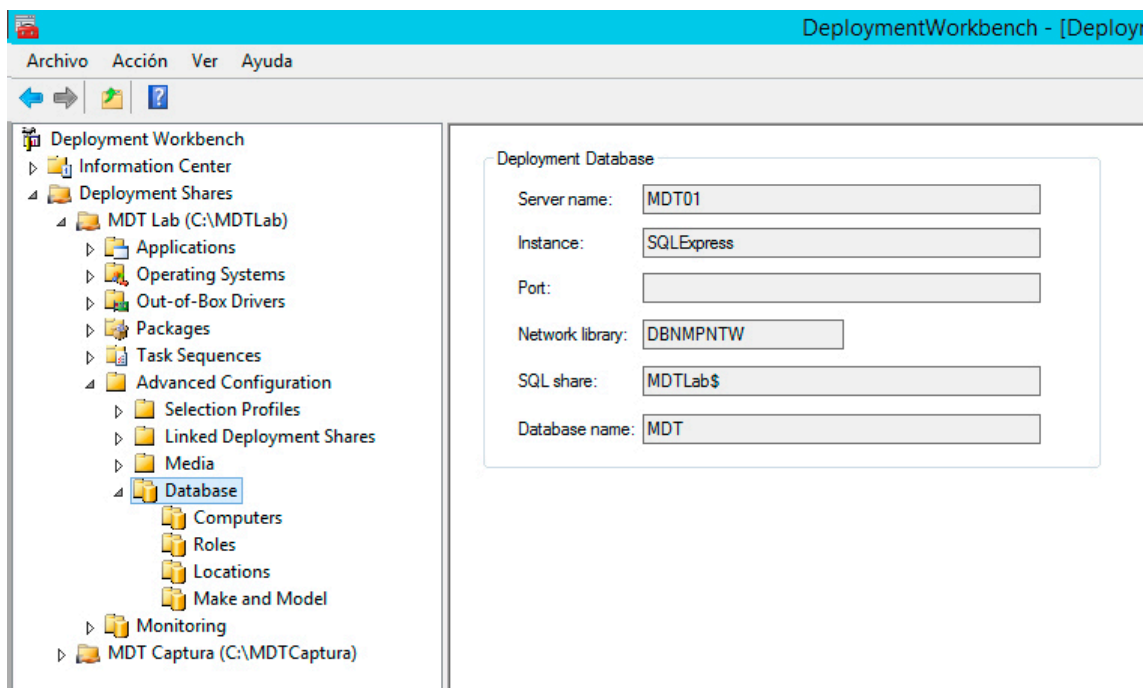


Figura 78: La base de datos MDT creada en el Deployment Workbench.

4.9.3.2. Permitir que el usuario UsuarioMDT pueda leer la base de datos de MDT

El siguiente paso es dar permisos de lectura al usuario (UsuarioMDT) en la base de datos de MDT, para que cuando se realizaran los despliegues, la secuencia de tareas pueda interrogar la base de datos en busca de los datos del ordenador. Para ello se ejecuta SQL Server Management Studio [29].

Expandimos la carpeta de Seguridad, y en Inicios de sesión, pulsando con el botón derecho del ratón seleccionamos Nuevo inicio de sesión.

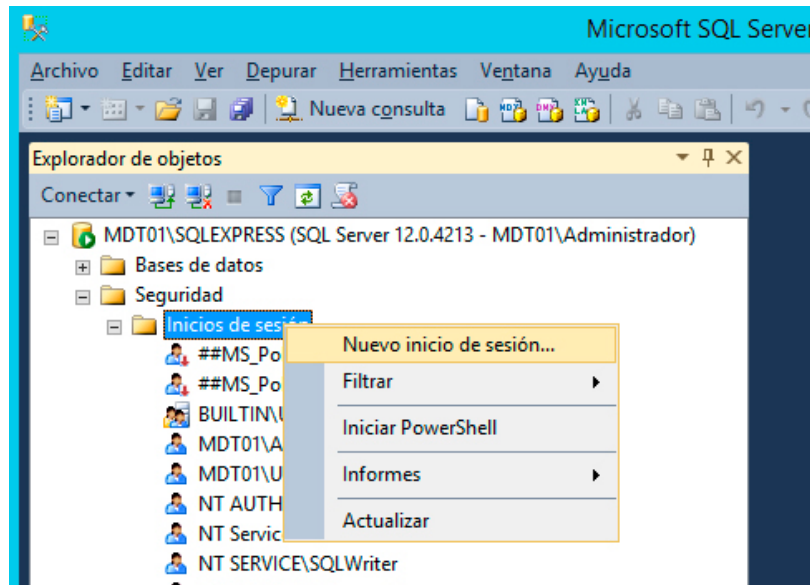


Figura 79: La carpeta de seguridad de la base de datos de MDT.

En la ventana de **Inicio de sesión – Nuevo**, se pulsa el botón Buscar y se localiza al usuario MDT01\UsuarioMDT. En el panel de la izquierda se selecciona Asignación de usuarios. Se marca la base de datos MDT, y se le asignan los siguientes roles:

1. db_datareader
2. public (por defecto)

Se pulsa Aceptar y se cierra SQL Server Management Studio.

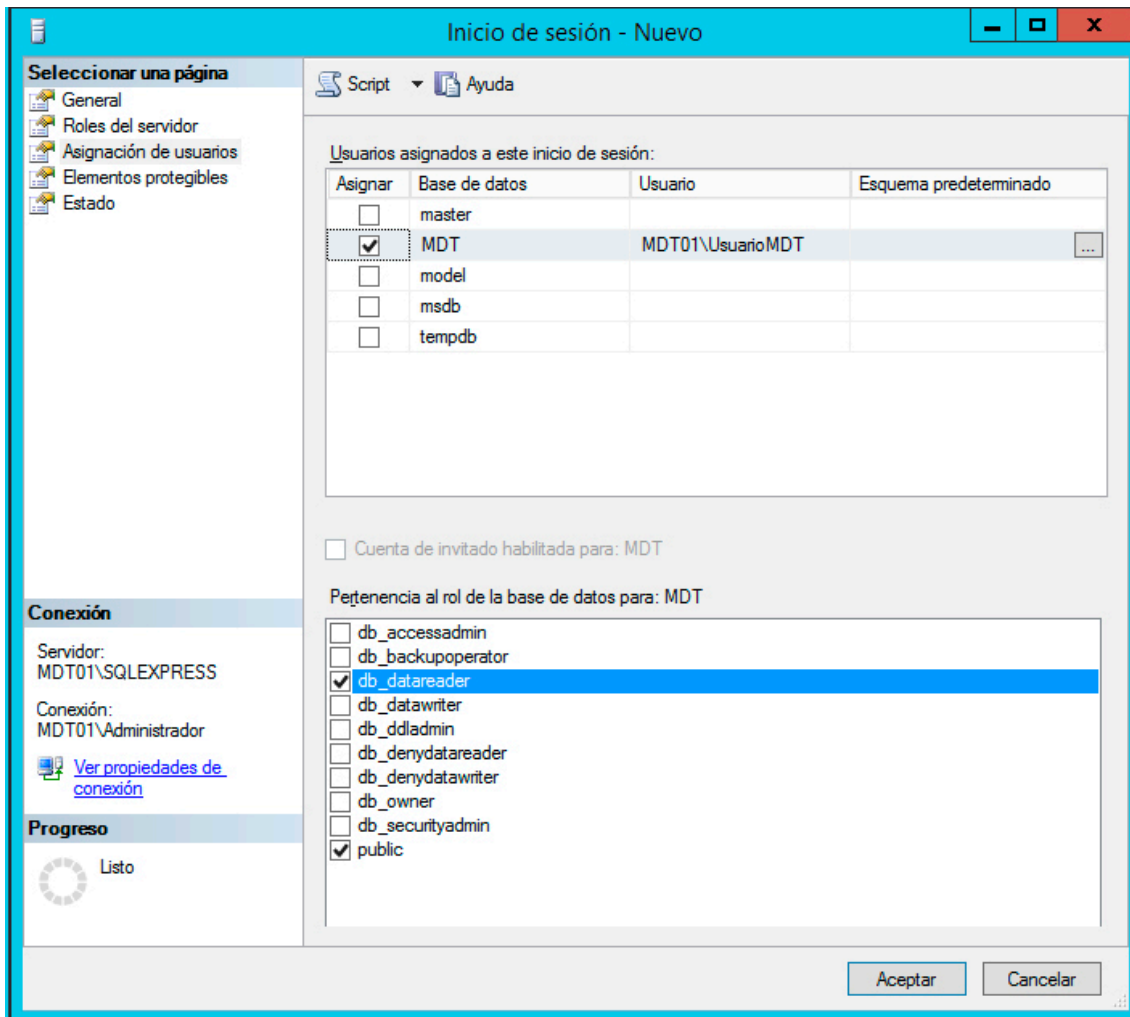


Figura 80: El nuevo inicio de sesión y permisos del UsuarioMDT en la BD de MDT.

4.9.3.3. Almacenar información de los ordenadores en la base de datos

A continuación, se detallan los pasos que se tienen que seguir para almacenar los datos de nuestros ordenadores en la base de datos de MDT.

4.9.3.3.1. Crear un nuevo registro de un ordenador

Para poder guardar la información de un ordenador en la base de datos se tiene que crear un nuevo registro. Dentro del DS MDTLab se selecciona la carpeta Advanced Configuration, Database, Computers, se pulsa el botón de la derecha del ratón y al marcar New nos muestra, un asistente que rellenamos con los siguientes datos:

1. **Description:** Ordenador de pruebas, imagen de Windows 10 REF01
2. **Asset tag:** WINDOWS10-REF01
3. **UUID:** 56D07648-0A46-11DE-BBDA-B31A52710025
4. **Serial number:** CZC94932LK
5. **MAC address:** 00:25:B3:1A:52:71

El ID, es un identificador que se incrementa automáticamente cada vez que se crea un nuevo registro, y no lo podemos manipular.



Figura 81: El nuevo ordenador creado en el Deployment Workbench.

4.9.3.3.2. Añadir información adicional para hacer despliegues ZTI

Ahora que ya se ha creado el registro del ordenador se puede completar la información del ordenador para hacer despliegues ZTI. Para ello se debe de acceder al registro del ordenador y se selecciona la pestaña Details. Aquí se muestran todas las propiedades del ordenador que se pueden rellenar para automatizar el despliegue. Gracias a la gran cantidad de parámetros configurables, se puede detallar hasta el más mínimo aspecto de la configuración de cada ordenador individualmente.

Hay varios grupos de propiedades:

- ADDS Settings: configuraciones de los servicios de directorio activo (AD), aplicable a servidores.
- Bitlocker: parámetros de Bitlocker.
- ConfigMgr 2012 OSD: configuración para despliegues hechos con SCCM.
- DHCP Server Settings: datos del servidor de DHCP, aplicable a servidores.
- Disk Settings: configuración de la unidad de disco del ordenador.
- Display Settings: configuración de la resolución de la pantalla.
- DNS Server Settings: datos de los DNS, aplicable a servidores.
- Domain and Workgroup: dominio o grupo donde se agregará el ordenador.
- Identification: identificación del ordenador.
- Miscellaneous: varias configuraciones.
- NIC Settings: configuración del adaptador de red.
- OS Roles: configuración de los roles del sistema operativo.
- Regional and Locale Settings: datos de configuración local y regional.
- User Data: configuración de la migración de los datos del usuario.
- Wizard Control: configuración del asistente de despliegue.
- Custom: parámetros personalizables.

Seguidamente se mostrarán los datos necesarios para poder realizar un despliegue ZTI para este caso. Para ello se rellenan las siguientes propiedades:

- en el grupo **Display Settings**: Como se conocen de antemano las características del monitor que utilizara el ordenador donde vamos a desplegar la imagen del sistema operativo, se introducen con los datos correctos del monitor.
 1. *BitsPerPixel*: 32 (número de bits por pixel).
 2. *Vrefresh*: 60 (velocidad de refresco vertical de la pantalla).
 3. *Xresolution*: 1920 (resolución horizontal).
 4. *Yresolution*: 1080 (resolución vertical).

Con estos datos, cuando la imagen se despliegue, nos aseguramos de que la imagen emitida en la pantalla se mostrara correctamente.
- en el grupo **Identification**:
 1. *OSDComputerName*: PruebaW10 (el nombre que se asigna al ordenador durante el despliegue de la imagen).
- en el grupo **Miscellaneous**:
 1. *OSInstall*: YES (para que se instale un sistema operativo).
 2. *TaskSequenceID*: REF01 (el identificador de la secuencia de tareas que se quiere ejecutar cuando se inicie un despliegue sobre esta máquina).
- en el grupo **Wizard Control**:
 1. *SkipComputerName*: YES (que no nos pregunte el nombre del ordenador, porque ya se lo hemos asignado en la propiedad *OSDComputerName* del grupo *Identification*).
 2. *SkipTaskSequence*: YES (que no nos pregunte que secuencia se desea desplegar, porque ya se la hemos asignado en la propiedad *TaskSequenceID* del grupo *Miscellaneous*).

Estos datos se graban en el registro del ordenador y después de configurar correctamente la base de datos, ya se pueden realizar despliegues ZTI.

4.9.3.4. Añadir aplicaciones para instalar en ordenadores concretos

Como ya hemos dicho anteriormente, se pueden añadir aplicaciones a ordenadores específicos que tengamos en la base de datos de MDT y que se instalaran al final del despliegue. En las aulas de informática tenemos varios ordenadores que tienen conectado un escáner de documentos. Como ejemplo se va a importar la aplicación de control del escáner y se añadirá al registro del ordenador que tiene conectado el escáner. Para ello se necesita conocer los parámetros de instalación desatendida del software.

En este caso se trataba de un fichero en formato MSI, y el parámetro de instalación desatendida es: /qb-, donde q es *quiet* (silencioso) y b- significa que no muestre el interfaz gráfico de la instalación. Como es un fichero MSI se ejecutará con el siguiente comando, donde /i, indica realizar la instalación:

```
msiexec /i "CanonScan.msi" /qb-
```



Se añade la aplicación al DS MDTLab con los siguientes datos (ver punto 4.8.1.):

1. Application Type

- Se selecciona: *Application with source files*

2. Details

- *Application Name: Canon CanonScan*

3. Source

- *Source directory: C:\CanonLide*
- No se selecciona: *Move the files to the deployment share instead of copying them*

4. Destination

- *Specify the name of the directory that should be created: Canon CanonScan*

5. Command Details

- *Command line: msiexec /i "CanonScan.msi" /qb-*
- *Working directory: .\Applications\Canon CanonScan*

Ahora hay que asignar la aplicación a un ordenador. Se pulsa en Advanced Configuration, Database, Computers, localizamos el registro del ordenador al que queríamos asignar la instalación de este programa, se accede a sus propiedades y en la pestaña Applications, se selecciona Add, Lite Touch Applications, y nos muestra todas las aplicaciones que tenemos el DS y que se pueden instalar. En nuestro caso se muestran las dos aplicaciones que existen en el DS:

- Antivirus Kaspersky
- Canon CanonScan

Se elige Canon CanonScan y ya se ha creado la asignación de este software para que se instale automáticamente al final del despliegue en el ordenador seleccionado. Ahora tenemos que modificar las reglas del DS para que interroguen a la base datos de MDT y saber que aplicaciones hay asociadas a cada ordenador cuando se realice el despliegue.

Se selecciona la carpeta Advanced Configuration, Database, pulsamos sobre botón derecho del ratón, marcamos Configure Database Rules, y seguimos el asistente:

En **Computer Options**, seleccionamos:

1. Query for computer specific settings
2. Query for applications to be installed on this computer

Se dejan desmarcadas el resto de opciones del asistente y finalizamos el asistente. Si miramos las reglas del DS, vemos que se ha añadido una nueva prioridad CApps, que interroga la tabla ComputerApplications, para saber que aplicaciones específicas tiene asignadas cada ordenador.

```

1 [Settings]
2 Priority=CApps, CSettings, Default
3 Properties=MyCustomProperty
4
5 [Default]
6 ; Instalar un sistema operativo = SI
7 OSInstall=Y
8
9 ; No preguntar el password del Administrador = SI
9 SkipAdminPassword=YES
...
66 EventService=http://MDT01:9800
67
68 [CSettings]
69 SQLServer=MDT01
70 Instance=SQLEXPRESS
71 Database=MDT
72 Netlib=DBNMPNTW
73 SQLShare=MDTLab$
74 Table=ComputerSettings
75 Parameters=UUID, AssetTag, SerialNumber, MacAddress
76 ParameterCondition=OR
77
78 [CApps]
79 SQLServer=MDT01
80 Instance=SQLEXPRESS
81 Database=MDT
82 Netlib=DBNMPNTW
83 SQLShare=MDTLab$
84 Table=ComputerApplications
85 Parameters=UUID, AssetTag, SerialNumber, MacAddress
86 ParameterCondition=OR
87 Order=Sequence

```

Código 5: Reglas para instalar aplicaciones en un ordenador específico.

4.9.4. Utilizar WDS para arrancar el ordenador utilizando PXE

Como ya se ha comentado en el punto 4.1., se puede utilizar WDS como herramienta complementaria de los despliegues de imágenes de MDT. Lo que nos interesa de WDS es su capacidad de poder utilizar el entorno PXE; que nos permitirá arrancar un ordenador vía red sin necesidad de un soporte físico.

4.9.4.1. Agregar el servicio WDS en Windows Server 2012

Se siguen los pasos del punto 4.5.2., pero esta vez se selecciona la opción de los servicios de implementación de Windows.

Se cumplimentan los siguientes datos u opciones en el asistente:

1. Opciones de instalación

- Servidor independiente. Aunque el ordenador está integrado en el dominio alumno de la UPV, no tenemos control sobre ese dominio.

2. Ubicación de carpeta de instalación remota

- Ruta: c:\RemoteInstall. Microsoft desaconseja el utilizar la misma unidad para albergar el sistema operativo las imágenes, pero como es para hacer pruebas realmente no importa.

3. Configuración inicial del servidor PXE

- No responder a ningún equipo cliente. Esta configuración se puede cambiar posteriormente, de momento, no queremos que este activo.

4. Operación completada

- Se desmarca la opción Agregar imágenes al servidor ahora.

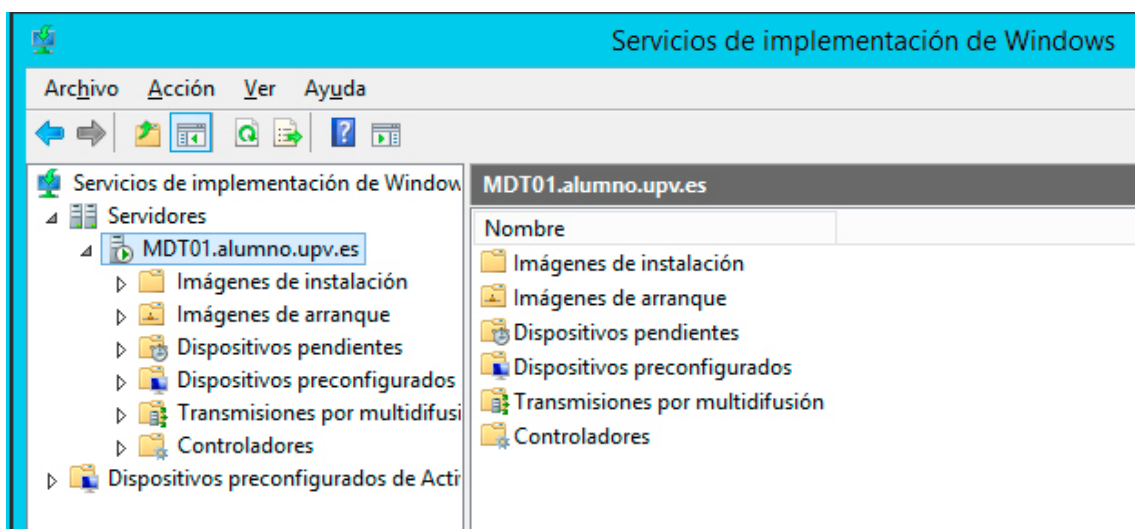


Figura 82: La consola de servicios de implementación de Windows.

4.9.4.2. Agregar imágenes de arranque de MDT

Para que el entorno PXE pueda funcionar y entregar una imagen de arranque de MDT, se deben de agregar las imágenes de arranque desde un fichero WIM. Las imágenes de arranque de MDT se crearon cuando se actualizó el DS MDTLab y las podíamos encontrar en la carpeta *Boot* dentro de la estructura de carpetas del DS. Allí se muestran dos ficheros WIM de arranque:

- LiteTouchPE_x86.wim. Esta imagen de arranque está basada en Windows PE con arquitectura de 32 bits, y tiene la particularidad de que puede desplegar imágenes de Windows con arquitecturas de 32 bits y de 64 bits.
- LiteTouchPE_x64.wim. Esta imagen de arranque está basada en Windows PE con arquitectura de 64 bits, y solo puede desplegar imágenes de Windows con arquitectura de 64 bits.

Se selecciona el servidor MDT01, la carpeta Imágenes de arranque, se pulsa el botón derecho del ratón y marcamos Agregar imagen de arranque. Se muestra un asistente, que se rellena con los siguientes datos:

1. Archivo de imagen, se pulso sobre el botón Examinar

- C:\MDTLab\Boot\ LiteTouchPE_x64.wim

2. Metadatos de imagen

En este paso podríamos dejar los datos que muestra por defecto, o bien, escribir algo que nos ayude a recordar a que DS hace referencia.

- Nombre de la imagen: MDTLab Lite Touch Windows PE (x64)
- Descripción de la imagen MDTLab Lite Touch Windows PE (x64)

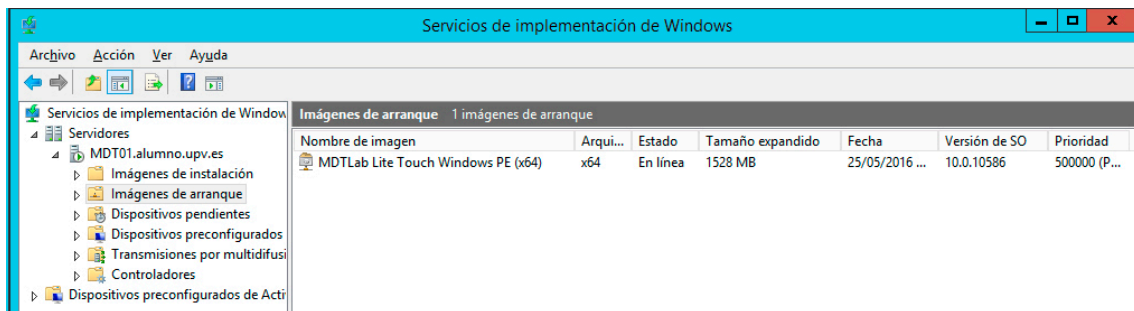


Figura 83: La imagen de arranque LiteTouchPE_x64.wim agregada a imágenes de arranque de WDS.

Ahora se podría repetir el asistente para agregar la imagen de arranque LiteTouchPE_x86.wim, pero como solo vamos a realizar despliegue de sistemas operativos de 64 bits, no es necesario que la importemos en WDS.

4.9.4.3. Seleccionar la imagen de arranque PXE por defecto

Si se muestran las propiedades del servidor MDT01 en la pestaña Arranque, se puede elegir la imagen predeterminada que se cargara en función de la arquitectura del procesador del ordenador. Se seleccionan las arquitecturas que vamos a utilizar y se dejan en blanco las que no vamos a utilizar.

Se seleccionan las siguientes opciones:

- Arquitectura x64: MDTLab Lite Touch Windows PE (x64)
- Arquitectura x64 (UEFI): MDTLab Lite Touch Windows PE (x64)

En las opciones de Directivas de arranque PXE, se pueden dejar las opciones por defecto, o cambiarlas para adecuarlas a nuestras necesidades. En nuestro caso lo que se quiere es interactuar lo mínimo con el despliegue, por lo que se selecciona lo siguiente:

Cientes conocidos: Continuar siempre con el arranque PXE

Cientes desconocidos: Continuar siempre con el arranque PXE

Hay que tener en cuenta que en la red donde se habilita el entorno PXE todos los clientes son conocidos, es decir; que cualquier cliente de la VLAN donde se encuentra el ordenador y el servidor MDT01, si solicita un arranque por PXE, se le entregara la imagen de arranque del DS MDTLab. Si no fuera ese el caso, habría que configurar correctamente las dos opciones anteriores.

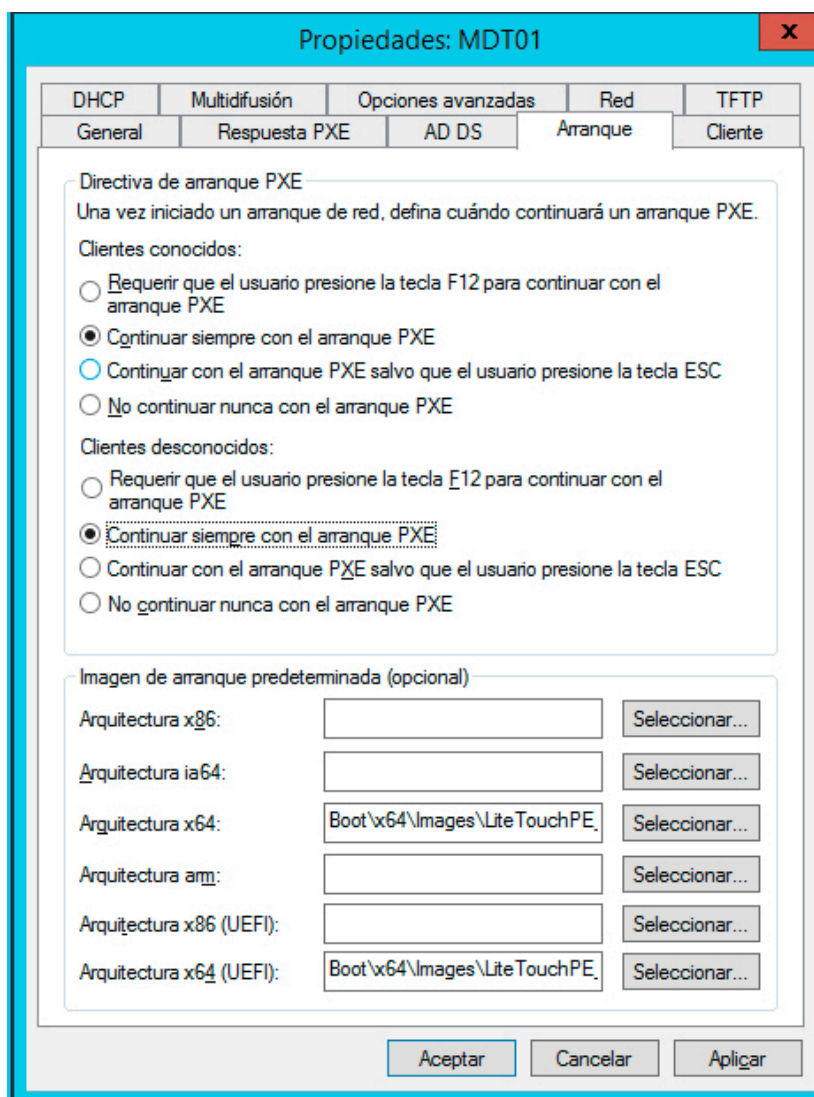


Figura 84: Las propiedades de arranque de WDS en el servidor MDT01.

4.9.4.4. Activar entorno PXE en WDS

Dentro de las propiedades del servidor MDT01, en la pestaña Respuesta PXE, se encuentra la configuración del entorno PXE. Al instalar el rol de WDS en el servidor, se ha dejado la directiva de respuesta como **No responder a ningún equipo cliente**. Ahora para poder habilitar la respuesta de PXE, se selecciona la opción:

- **Responder a todos los equipos cliente (conocidos y desconocidos)**

En nuestro caso se deja desmarcada la opción **Requerir aprobación del administrador para los equipos desconocidos**. La razón para ello es porque cualquier ordenador que solicite un arranque PXE al servidor MDT01 se encuentra dentro de la misma VLAN y es un ordenador gestionado por nosotros.

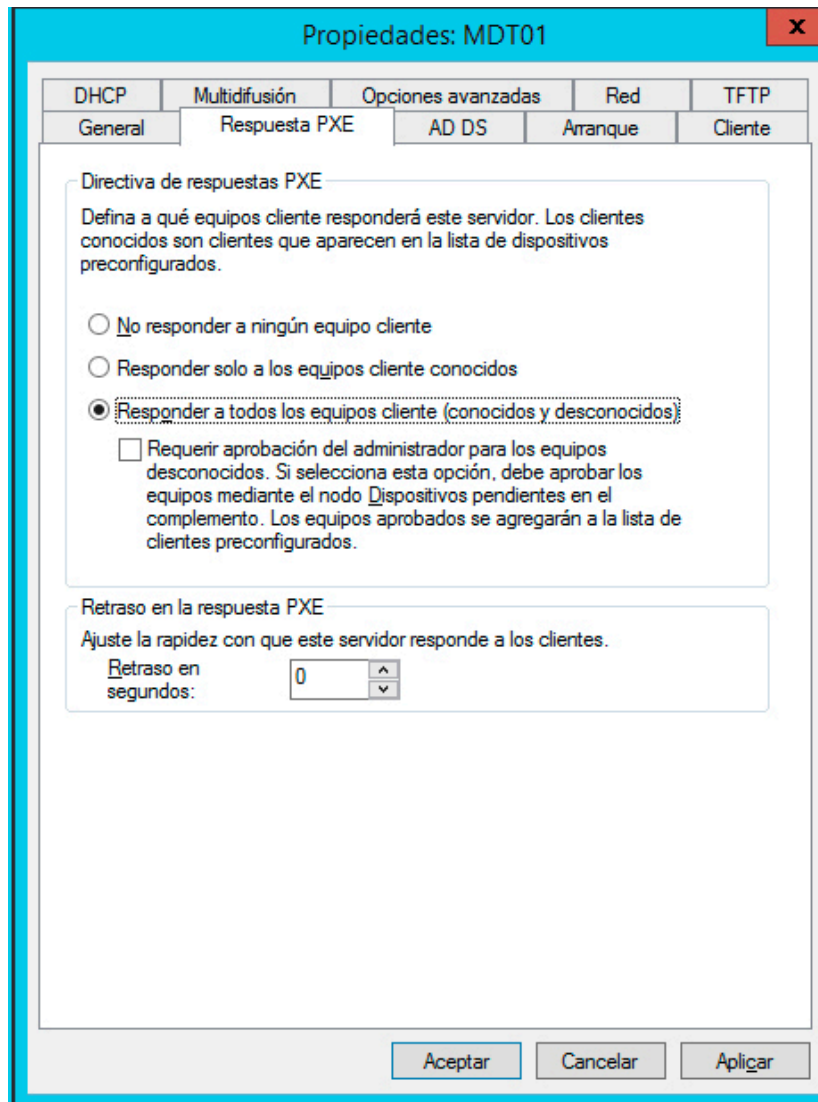


Figura 85: Las propiedades de respuesta PXE de WDS en el servidor MDT01.

4.9.5. Como iniciar un despliegue

Hasta ahora habíamos visto una única manera de iniciar un despliegue, utilizando un medio físico, bien un CD/DVD o una UFD. Existen otras dos maneras de hacerlo para llegar a una automatización total.

4.9.5.1. Iniciar despliegue utilizando un medio físico

Este método ya se ha comentado y consiste en grabar en un CD/DVD o una UFD, el fichero ISO de arranque del DS MDTLab. Como ya se ha dicho nosotros únicamente realizamos despliegues de sistemas operativos de 64 bits, y por ello se utiliza el fichero LiteTouchPE_x64.iso, que se graba en un medio físico. Después de arrancar el ordenador con el medio físico, el ordenador se conectará con el DS MDTLab y comenzará el despliegue.

4.9.5.2. Iniciar despliegue utilizando PXE

Para poder hacer despliegues utilizando PXE, se necesita conocer la tecla de acceso al arranque PXE del ordenador. Cuando el ordenador se enciende, en el proceso de arranque del BIOS, nos suele indicar cuál es la tecla de acceso al arranque PXE. La mayoría de los fabricantes utilizan la tecla de función 12 [**F12**], y se identifica con una leyenda que se muestra en la pantalla, como PXE Boot o Network Boot.

Pulsando la tecla [**F12**] cuando el ordenador está arrancando, este se comunicará con el entorno de PXE de los servicios de implementación de Windows; descargará la imagen por defecto para la arquitectura de 64 bits (MDTLab Lite Touch Windows PE (x64)), e iniciará el despliegue automáticamente.

4.9.5.3. Iniciar despliegue ejecutando el fichero LiteTouch.vbs desde el DS MDTLab

Si se ejecuta el fichero LiteTouch.vbs desde un sistema operativo en ejecución, MDT entiende que lo que se quiere hacer es una actualización del sistema o una reinstalación del mismo. Estas dos opciones suponen que MDT realizará una copia de la configuración y de los datos del usuario antes del despliegue, y cuando este finalice, devolverá la información de la configuración y de los datos del usuario al ordenador. En las aulas de uso público, no se desea guardar ningún dato ni configuración de los numerosos usuarios que existen en el ordenador. Las siguientes líneas de las reglas del DS MDTLab, que se localizan dentro del grupo [Default] hacen que no se realice la copia de seguridad de los datos del usuario.

```
...
12 ; No preguntar el hacer un copia de seguridad del ordenador = SI
13 SkipComputerBackup=YES
...
20 ; No preguntar por guardar y recuperar los datos del usuario = SI
21 SkipUserData=YES
22 ; Ubicacion para los ficheros de datos del usuario = NINGUNA
23 UserDataLocation=NONE
...
```

Código 6: Líneas de Rules para no hacer copia de seguridad de los datos de usuario.

Desde el ordenador donde queremos que reciba el despliegue de una imagen, tenemos que realizar una conexión con el recurso compartido MDTLab\$ del servidor MDT01 para poder ejecutar el fichero LiteTouch.vbs. Si se utiliza un usuario que no tenga credenciales para conectarse con el recurso compartido, Windows nos solicitará unas nuevas credenciales para conectarse con dicho recurso. Al ejecutar LiteTouch.vbs se van a realizar cambios en el ordenador, por lo que se solicita permiso de administrador para su ejecución. Debido a esto, el despliegue de la imagen no es un proceso totalmente automático.

4.9.6. Conseguir despliegue automatizado

Como se ha visto en el punto 3.2., hay otras herramientas que permiten realizar Wake-On-Lan de los ordenadores y desplegar imágenes de modo desatendido. Se puede alcanzar el mismo grado de automatización usando MDT. Para conseguirlo, se necesita:

1. Crear una tarea programada en los ordenadores clientes.
2. Crear un *script* de Windows Powershell que gestiona el proceso de despliegue.
3. Crear una tarea programada en el servidor MDT01, que lanza el *script* creado en el paso anterior.

4.9.6.1. Crear una tarea programada en el ordenador de referencia

Se puede conseguir que los clientes tengan una tarea programada para realizar despliegues de dos maneras:

1. Creando la tarea programada en el ordenador de referencia de donde se leerá posteriormente la imagen de referencia.
2. Suponiendo que los ordenadores están integrados en un dominio de AD, y utilizando una política de grupo, o GPO, para crear la tarea después del despliegue.

Se mostrará cómo crear la tarea programada en el ordenador de referencia (aunque el proceso utilizando una GPO es similar). En el ordenador de referencia se tiene que crear una tarea programada con las siguientes características:

1. General

- **Nombre:** DespliegueMDTLab
- Se selecciona: **Ejecutar solo cuando el usuario haya iniciado sesión.** Este punto es importante, debe de haber una sesión iniciada para que se pueda hacer el despliegue totalmente automatizado. Esto se debe a la necesidad de tener un usuario con credenciales suficientes para conectarse al recurso de red MDTLab\$ del servidor MDT01.
- Se selecciona: **Ejecutar con los privilegios más altos**
- **Configurar para:** Windows 10



2. Acciones

- Acción: **Iniciar un programa**
- **Programa o script:** \\mdt01\mdtlab\$\scripts\LiteTouch.vbs

3. Configuración

- Se selecciona: **Permitir que la tarea se ejecute a petición**

4.9.6.2. Script para realizar despliegues totalmente automáticos

Vamos a crear un *script* en Windows Powershell que realiza los despliegues automáticamente, sin ninguna intervención. Los pasos a seguir son:

- Leer de un fichero XML los nombres de los ordenadores que intervendrán en el despliegue.
- Encender remotamente cada uno de los ordenadores mediante Wake-On-Lan, utilizando los datos guardados en la base de datos de MDT.
- Como no todos los ordenadores arrancar con la misma rapidez, se crea un hilo por cada ordenador y que se encargará de lo siguiente:
 1. Esperar un tiempo suficiente para que el ordenador solicite el *login* del usuario.
 2. Abrir una sesión de escritorio remoto (RDP) en el ordenador con el usuario adminMDT.
 3. Eliminar el proceso de sesión de RDP dejando la sesión iniciada.
- Ejecutar remotamente la tarea programada DespliegueMDTlab en el ordenador cliente.

El *script* se llama **DespliegueAutomatico.PS1**, y lee de un fichero **ORDENADORES_DESPLIEGUE.xml** los nombres de los ordenadores que van a participar en el despliegue.

```
1 # Creado por Miguel Gómez Sabater
2 # Durante los años 2015 y 2016
3
4 # Se añade System.Data para
5 # poder llamar a las funciones de SQL Server
6 Add-Type -AssemblyName System.Data
7
8 # INICIO DE LAS FUNCIONES Y SCRIPTBLOCKS
9
10 # INICIO DE LA FUNCION LOG
11 # el parametro $Que es lo que queremos grabar en el log
12 function Log($Que)
13 {
14 # recuperamos el nombre de la maquina que hace el clonado
15 $Maquina = $env:ComputerName
16 # asignamos el directorio para guardar los logs
17 $DirectorioLOG = "."
18 # fichero de log del propio programa
19 $FicheroLOG = $DirectorioLOG + "\Despliegue_" + $Maquina + ".log"
20 # recuperamos la fecha actual
```

```

21 $MiFecha = Get-Date -format d
22 # recuperamos la hora actual
23 $MiHora = Get-Date -format HH:mm:ss
24 # se graba en el fichero log
25 $MiFecha + " " + $MiHora + " " + $Que | Out-File -Append -FilePath
$FicheroLOG
26 }
27 # FIN DE LA FUNCION LOG
28
29 # INICIO DE LA FUNCION WOL
30 # funcion que realiza WOL (Wake On Lan) de los ordenadores
31 # Como en la base de datos de MDT tenemos lo datos del nombre del
ordendaor y su mac
32 # se pasa como parametro el nombre del ordenador y buscamos en la base
datos su registro
33 # para conocer el MAC y lanzar un par de paquetes WOL
34 Function WOL([string]$miPC)
35 {
36 # cadena de conexión con el servidor de SQL donde esta la base de
datos de MDT
37 $mdtSQLConnectionString = "Server='CLONATORMN'; Database='MDT';
Integrated Security=true;"
38 # se crea un nuevo objeto de conexion con SQL
39 $mdtSQLConnection = new-object System.Data.SqlClient.SqlConnection
40 $mdtSQLConnection.ConnectionString = $mdtSQLConnectionString
41 # se abre la conexión
42 $mdtSQLConnection.Open()
43 # sentencia SQL para recuperar el registro correspondiente a un
ordenador
44 $sql = "SELECT * FROM ComputerSettings WHERE AssetTag='" + $miPC +
""
45 $selectAdapter = New-Object System.Data.SqlClient.
SqlDataAdapter($sql, $mdtSQLConnection)
46 $selectDataset = New-Object System.Data.Dataset
47 $null = $selectAdapter.Fill($selectDataset, "ComputerSettings")
48 # se asigna a $PC la fila recuperada del select de SQL
49 $PC = $selectDataset.Tables[0].Rows
50 # recuperamos la dirección MAC del PC
51 $MAC = $PC.MacAddress
52
53 # se envian dos paquetes de WOL a la dirección MAC del ordenador
54 for ($i = 1; $i -le 2; $i++)
55 {
56 # el siguiente código es de Kris Powell y esta publicado en línea
en
57 # http://www.adminarsenal.com/admin-arsenal-blog/powershell-
sending-a-wake-on-lan-wol-magic-packet/
58 $MacByteArray = $MAC -split "[:-]" | ForEach-Object { [Byte] "0x$_"
}
59 [Byte[]]$MagicPacket = ( , 0xFF * 6) + ($MacByteArray * 16)
60 $UdpClient = New-Object System.Net.Sockets.UdpClient
61 $UdpClient.Connect(([System.Net.IPAddress]::Broadcast), 9)
62 $UdpClient.Send($MagicPacket, $MagicPacket.Length) | Out-Null
63 $UdpClient.Close()
64 # esperamos un segundo para lanzar el segundo paquete WOL
65 Start-Sleep -Seconds 1
66 }
67

```



```

68 }
69 # FIN DE LA FUNCION WOL
70
71 # INICIO DEL SCRIPTBLOCK DE LOGIN
72 # bloque de codigo que se lanza para cada maquina para realizar el
login
73 # cada maquina tendra un job que se encargara de ella
74 $ScriptBlockLogin = {
75 param ($Computer,
76 $TimeOUT)
77
78 Function Connect-RDP
79 {
80 Param
81 (
82 [Parameter(Mandatory = $true,
83 ValueFromPipelineByPropertyName = $true,
84 Position = 0)]
85 $Computer
86 )
87 # inicializamos un contador de intentos
88 $Contador = 1
89 do
90 {
91 # se intenta conectar con el protocolo de RDP del ordenador
cliente
92 $ConTest = (Test-NetConnection -ComputerName $Computer
-CommonTCPPort RDP).TcpTestSucceeded
93 # por cada intento de conectar con el ordenador incrementamos
en uno el contador
94 $Contador++
95 }
96 # finalmente, si no hay conectividad despues de 5 intentos,
salimos del bucle
97 until ($ConTest -eq $true -or $Contador -eq 5)
98 # se comprueba si hay conectividad con el ordenador para hacer
RDP
99 if ($ConTest -eq $true)
100 {
101 # si hay conectividad se lanza la sesion de RDP
102 Connect-Mstsc -ComputerName $Computer -User alumno\adminMDT
-Password password -Width 800 -Height 600
103 # se pasa el $TimeOUT como un parametro, leido de .\
ORDENADORES_DESPLIEGUE.xml
104 $Esperar = $TimeOUT
105
106 # bucle de espera
107 for ($i = 1; $i -le $Esperar; $i++)
108 {
109 $Segundos = $Esperar - $i
110 Start-Sleep 1
111 }
112
113 # se localiza el proceso de escritorio remoto que hemos
iniciado para este ordenador
114 $Proceso = $Computer + " - Conexión a Escritorio remoto"
115 # se elimina el proceso, pero deja abierta la sesion de RDP
116 Get-Process | Where-Object { $_.MainWindowTitle -eq $Proceso }

```

```
| stop-process
117     }
118 }
119
120 # la siguiente funcion es de Jaap Brassler
121 # link: http://www.jaapbrasser.com
122 function Connect-Mstsc
123 {
124     <#
125     .SYNOPSIS
126     Function to connect an RDP session without the password prompt
127
128     .DESCRIPTION
129     This function provides the functionality to start an RDP session
130     without having to type in the password
131
132     .PARAMETER ComputerName
133     This can be a single computername or an array of computers to
134     which RDP session will be opened
135
136     .PARAMETER User
137     The user name that will be used to authenticate
138
139     .PARAMETER Password
140     The password that will be used to authenticate
141
142     .PARAMETER Credential
143     The PowerShell credential object that will be used to
144     authenticate against the remote system
145
146     .PARAMETER Admin
147     Sets the /admin switch on the mstsc command: Connects you to the
148     session for administering a server
149
150     .PARAMETER MultiMon
151     Sets the /multimon switch on the mstsc command: Configures the
152     Remote Desktop Services session monitor layout to be identical to the
153     current client-side configuration
154
155     .PARAMETER FullScreen
156     Sets the /f switch on the mstsc command: Starts Remote Desktop in
157     full-screen mode
158
159     .PARAMETER Public
160     Sets the /public switch on the mstsc command: Runs Remote Desktop
161     in public mode
162
163     .PARAMETER Width
164     Sets the /w:<width> parameter on the mstsc command: Specifies the
165     width of the Remote Desktop window
166
167     .PARAMETER Height
168     Sets the /h:<height> parameter on the mstsc command: Specifies the
169     height of the Remote Desktop window
170
171     .NOTES
172     Name: Connect-Mstsc
173     Author: Jaap Brassler
```



```
164     DateUpdated: 2016-01-12
165     Version: 1.2.2
166     Blog: http://www.jaapbrasser.com
167
168     .LINK
169     http://www.jaapbrasser.com
170
171     .EXAMPLE
172     . .\Connect-Mstsc.ps1
173
174     Description
175     -----
176     This command dot sources the script to ensure the Connect-Mstsc
177     function is available in your current PowerShell session
178
179     .EXAMPLE
180     Connect-Mstsc -ComputerName server01 -User contoso\jaapbrasser
181     -Password supersecretpw
182
183     Description
184     -----
185     A remote desktop session to server01 will be created using the
186     credentials of contoso\jaapbrasser
187
188     .EXAMPLE
189     Connect-Mstsc server01,server02 contoso\jaapbrasser supersecretpw
190
191     Description
192     -----
193     Two RDP sessions to server01 and server02 will be created using
194     the credentials of contoso\jaapbrasser
195
196     .EXAMPLE
197     server01,server02 | Connect-Mstsc -User contoso\jaapbrasser
198     -Password supersecretpw -Width 1280 -Height 720
199
200     Description
201     -----
202     Two RDP sessions to server01 and server02 will be created using
203     the credentials of contoso\jaapbrasser and both session will be at a
204     resolution of 1280x720.
205
206     .EXAMPLE
207     Connect-Mstsc -ComputerName server01:3389 -User contoso\
208     jaapbrasser -Password supersecretpw -Admin -MultiMon
209
210     Description
```

```

210      -----
211      A RDP session to server01 at port 3389 will be created using the
credentials of contoso\jaapbrasser and the /admin and /multimon switches
will be set for mstsc
212
213      .EXAMPLE
214      Connect-Mstsc -ComputerName server01:3389 -User contoso\
jaapbrasser -Password supersecretpw -Public
215
216      Description
217      -----
218      A RDP session to server01 at port 3389 will be created using the
credentials of contoso\jaapbrasser and the /public switches will be set
for mstsc
219
220      .EXAMPLE
221      Connect-Mstsc -ComputerName 192.168.1.10 -Credential $Cred
222
223      Description
224      -----
225      A RDP session to the system at 192.168.1.10 will be created using
the credentials stored in the $cred variable.
226
227      .EXAMPLE
228      Get-AzureVM | Get-AzureEndPoint -Name 'Remote Desktop' | ForEach-
Object { Connect-Mstsc -ComputerName ($_.Vip,$_.Port -join ':') -User
contoso\jaapbrasser -Password supersecretpw }
229
230      Description
231      -----
232      A RDP session is started for each Azure Virtual Machine with the
user contoso\jaapbrasser and password supersecretpw
233
234      .EXAMPLE
235      PowerShell.exe -Command "& { . .\Connect-Mstsc.ps1; Connect-Mstsc
server01 contoso\jaapbrasser supersecretpw -Admin}"
236
237      Description
238      -----
239      An remote desktop session to server01 will be created using
the credentials of contoso\jaapbrasser connecting to the administrative
session, this example can be used when scheduling tasks or for batch files.
240      #>
241      [cmdletbinding(SupportsShouldProcess, DefaultParametersetName =
'UserPassword')]
242      param (
243          [Parameter(Mandatory = $true,
244                      ValueFromPipeline = $true,
245                      ValueFromPipelineByPropertyName = $true,
246                      Position = 0)]
247          [Alias('CN')]
248          [string[]]$ComputerName,
249          [Parameter(ParameterSetName = 'UserPassword', Mandatory = $true,
250                      Position = 1)]
251          [Alias('U')]
252          [string]$User,
253          [Parameter(ParameterSetName = 'UserPassword', Mandatory = $true,
254                      Position = 2)]

```



```
253     [Alias('P')]
254     [string]$Password,
255     [Parameter(ParameterSetName = 'Credential', Mandatory = $true,
Position = 1)]
256     [Alias('C')]
257     [PSCredential]$Credential,
258     [Alias('A')]
259     [switch]$Admin,
260     [Alias('MM')]
261     [switch]$MultiMon,
262     [Alias('F')]
263     [switch]$FullScreen,
264     [Alias('Pu')]
265     [switch]$Public,
266     [Alias('W')]
267     [int]$Width,
268     [Alias('H')]
269     [int]$Height,
270     [Alias('WT')]
271     [switch]$Wait
272 )
273
274 begin
275 {
276     [string]$MstscArguments = ''
277     switch ($true)
278     {
279         { $Admin } { $MstscArguments += '/admin ' }
280         { $MultiMon } { $MstscArguments += '/multimon ' }
281         { $FullScreen } { $MstscArguments += '/f ' }
282         { $Public } { $MstscArguments += '/public ' }
283         { $Width } { $MstscArguments += "/w:$Width " }
284         { $Height } { $MstscArguments += "/h:$Height " }
285     }
286
287     if ($Credential)
288     {
289         $User = $Credential.UserName
290         $Password = $Credential.GetNetworkCredential().Password
291     }
292 }
293 process
294 {
295     foreach ($Computer in $ComputerName)
296     {
297         $ProcessInfo = New-Object System.Diagnostics.ProcessStartInfo
298
299         $Process = New-Object System.Diagnostics.Process
300
301         # Remove the port number for CmdKey otherwise credentials are
not entered correctly
302         if ($Computer.Contains(':'))
303         {
304             $ComputerCmdkey = ($Computer -split ':')[0]
305         }
306         else
307         {
308             $ComputerCmdkey = $Computer
309         }
310     }
311 }
```



```

309
310     $ProcessInfo.FileName = "$($env:SystemRoot)\system32\cmdkey.
exe"
311     $ProcessInfo.Arguments = "/generic:TERMSRV/$ComputerCmdkey /
user:$User /pass:$Password"
312     $ProcessInfo.WindowStyle = [System.Diagnostics.
ProcessWindowStyle]::Hidden
313     $Process.StartInfo = $ProcessInfo
314     if ($PSCmdlet.ShouldProcess($ComputerCmdkey, 'Adding
credentials to store'))
315     {
316         [void]$Process.Start()
317     }
318
319     $ProcessInfo.FileName = "$($env:SystemRoot)\system32\mstsc.
exe"
320     $ProcessInfo.Arguments = "$MstscArguments /v $Computer"
321     $ProcessInfo.WindowStyle = [System.Diagnostics.
ProcessWindowStyle]::Normal
322     $Process.StartInfo = $ProcessInfo
323     if ($PSCmdlet.ShouldProcess($Computer, 'Connecting mstsc'))
324     {
325         [void]$Process.Start()
326         if ($Wait)
327         {
328             $null = $Process.WaitForExit()
329         }
330     }
331 }
332 }
333 }
334
335 # inicio del codigo
336 $DirectorioLOG = "."
337 Connect-RDP $Computer
338
339 } # FIN DEL SCRIPTBLOCK DE LOGIN
340
341 # FIN DE LAS FUNCIONES Y SCRIPTBLOCKS
342
343 # INICIO DEL SCRIPT PRINCIPAL
344 $Version = "v1.2 25-5-2016"
345 # carpeta donde guardaremos el log de ejecución del script
346 $DirectorioLOG = "."
347
348 # Eliminamos los trabajos que existan
349 Get-Job | Remove-Job -Force
350
351 # Comienza el log del programa
352 $Mensaje= "======"
353 Log($Mensaje)
354 Write-Host $Mensaje
355 $Mensaje = "Lanzando despliegue desatendido " + $Version
356 Log($Mensaje)
357 Write-Host $Mensaje
358
359 # se comprueba si existe el fichero ORDENADORES_DESPLIEGUE.xml en
360 # la carpeta donde se esta ejecutando el script

```



```
361 $Mensaje = "Leyendo el fichero ORDENADORES_DESPLIEGUE.xml"
362 Log($Mensaje)
363 Write-Host $Mensaje
364
365 if (Test-Path ".\ORDENADORES_DESPLIEGUE.xml")
366 {
367     # si existe, asignamos a $Path el valor de la ruta del fichero
368     $Path = ".\ORDENADORES_DESPLIEGUE.xml"
369
370     # se crea un nuevo objeto XML
371     $xml = New-Object -TypeName XML
372     # se carga el fichero XML
373     $xml.Load($Path)
374
375     # leemos del nodo Parametros, el valor de tiempo de espera para se
realice
376     # el proceso de login en el ordenador
377     $Parametros = $Xml.Clonar.Parametros | Select-Object -Property *
378     [int]$TimeOUT = $Parametros.TiempoEsperaLogin
379
380     # contamos cuantos ordenadores que hay en el XML, que es el numero
de nodos
381     # seleccionamos los nodos
382     $Ordenadores = $xml.selectnodes("//Clonar/Ordenadores/Ordenador")
383     # numero de nodos, que es numero de PC que hay que desplegar la
imagen
384     $Num_PC = $Ordenadores.count
385
386     # si el numero de ordenadores es cero, es que hay algún problema
387     if ($Num_PC -eq 0)
388     {
389         # se escribe el error en el log y en la salida de consola del
programa
390         $Mensaje = "ERROR: No hay ningun PC en el fichero. Finalizando."
391         Log($Mensaje)
392         Write-Host $Mensaje
393         Exit
394     }
395
396     # escribimos los parametros en el log y en la consola de salida del
programa
397     $Mensaje = "Número de PCs en fichero es " + $Num_PC
398     Log($Mensaje)
399     Write-Host $Mensaje
400     $Mensaje = "Tiempo de espera para el Login es " + $TimeOUT + "
segundos."
401     Log($Mensaje)
402
403     Write-Host $Mensaje
404
405     # le restamos uno, porque los indices de los nodos del fichero XML
comienzan en cero
406     $Num_PC = $Num_PC - 1
407 } #if (Test-Path .\ORDENADORES_DESPLIEGUE.xml)
408 else
409 {
410     # no existe el fichero .\ORDENADORES_DESPLIEGUE.xml y el script no
```

```

puede continuar
411 $Mensaje = "ERROR: No existe el fichero .\ORDENADORES_DESPLIEGUE.xml
Finalizando."
412 Log($Mensaje)
413 Write-Host $Mensaje
414 # finalizamos el script
415 Exit
416 }
417
418 # bucle de ordenadores leidos del fichero XML
419 for ($z = 0; $z -le $Num_PC; $z++)
420 {
421 # recuperamos el nombre del ordenador del nodo actual del XML
422 [string]$computer = $Ordenadores.Item($z).Nombre
423 $Mensaje = "Despertando a " + $computer
424 # se escribe en el log y en la salida de consola del programa
425 Log($Mensaje)
426 Write-Host $Mensaje
427 # llamamos a la función que hace WOL de los ordenadores
428 WOL($computer)
429
430 # le añadimos LOGIN_ al nombre del ordenador/job
431 $NombreTarea = "LOGIN_" + $Computer
432 # se ejecuta el scriptblock que se encarga del login
433 Start-Job -scriptblock $ScriptBlockLogin -ArgumentList @($StartDate,
$computer, $TimeOUT) -Name $NombreTarea | Out-Null
434 } # for ($z = 0; $z -le $Num_PC; $z++)
435
436 # bucle de comprobación de que han terminado todos los jobs encargados
de hacer el login en los ordenadores
437 $NumeroJobsRunning = $(Get-Job -State Running).count
438 While ($(Get-Job -State Running | Where-Object { $_.Name.
Contains("LOGIN_") }).count -gt 0)
439 {
440 # se recupera el número de job pendientes y que estan en ejecución
441 $i = $(Get-Job -State Running | Where-Object { $_.Name.
Contains("LOGIN_") }).count
442
443 # escribimos el dato en la consola de salida del script
444 Write-Host $i LOGIN siguen en marcha -NoNewLine
445
446 $ComputersStillRunning = ""
447 ForEach ($System in $(Get-Job -State Running | Where-Object {
$_.Name.Contains("LOGIN_") }))) { $ ComputersStillRunning += ", $($System.
name)" }
448 Write-Host $ComputersStillRunning
449
450 if ($i -ne $NumeroJobsRunning)
451 {
452 Log("$i LOGIN siguen en marcha" + $ComputersStillRunning)
453 $NumeroJobsRunning = $i
454 }
455 # esperamos de 5 segundos
456 start-sleep -seconds 5
457 } #While ($(Get-Job -State Running | Where-Object { $_.Name.
Contains("LOGIN_") }).count -gt 0)
458
459 # bucle que ejecuta la tarea remota Clonar en el ordenador de destino

```



```
460 for ($z = 0; $z -le $Num_PC; $z++)
461 {
462     # recuperamos el nombre del ordenador
463     [string]$computer = $Ordenadores.Item($z).Nombre
464
465     # aqui se comprueba que hay una sesion iniciada de adminMDT, sino
    fallara
466     # https://technet.microsoft.com/es-es/library/cc731503%28v=ws.10%29.
    aspx
467     # Comando qwinsta: Muestra información acerca de las sesiones en un
    servidor Host de sesión de escritorio remoto (Host de sesión de escritorio
    remoto).
468     $RDP_Answer = (qwinsta /server:$computer | foreach { (($_trim()
    -replace "\s+", ",")) } | select-string "adminMDT")
469     # escribimos el resultado de comprobar si existe una sesión iniciada
    del usuario
470     Write-Host $RDP_Answer
471     # si la respuesta no es nula o vacia, es que existe una sesión
    iniciada
472     if (-not [string]::IsNullOrEmpty($RDP_Answer))
473     {
474         # se escribe en el log y en la salida de consola del programa
475         $Mensaje = "Intentando lanzar la tarea remota DesplegarMDTLab en "
    + $Computer
476         Log($Mensaje)
477         Write-Host $Mensaje
478
479         # se intenta ejecutar la tarea remota DesplegarMDTLab
480         # https://technet.microsoft.com/es-es/library/
    cc725744%28v=ws.10%29.aspx
481         # Comando SHTASKS: Programa comandos y programas para ejecutar
    periódicamente o en un momento determinado.
482         # Agrega y quita las tareas de la programación, se inicia y
    detiene tareas a petición y muestra y cambia tareas programadas.
483         $OutputVariable = schtasks.exe /run /s \\$computer /tn
    DespliegueMDTLab 2>&1 | Out-String
484         # se escribe el resultado de schtasks en el log y en la salida de
    consola del programa
485         Log($OutputVariable)
486         Write-Host $OutputVariable
487
488         $Mensaje = "Lanzada tarea remota DespliegueMDTLab en " + $computer
489         Log($Mensaje)
490         Write-Host $Mensaje
491     }
492     else
493     {
494         #no hay ninguna sesión abierta
495         #se escribe el error en el log y en la salida de consola del
    programa
496         $Mensaje = "ERROR: No HAY una sesión de adminMDT abierta en " +
    $Computer
497         Log($Mensaje)
498         Write-Host $Mensaje
499
500         $Mensaje = "ERROR: NO SE HA EJECUTADO la tarea programada
    DespliegueMDTLab en " + $Computer
501         Log($Mensaje)
502         Write-Host $Mensaje
503     }
504 }
```

```

502 }
503 } #for ($z = 0; $z -le $Num_PC; $z++)
504 # FIN DEL SCRIPT
505 $Mensaje = "FIN de la ejecución del despliegue"
506 Log($Mensaje)
507 Write-Host $Mensaje

```

Código 7: Contenido del fichero DespliegueAutomatico.PS1.

El *script* de despliegue automático necesita un fichero XML, que hemos llamado ORDENADORES_DESPLIEGUE.XML, donde se encuentran:

- un parámetro de tiempo en segundos, que es el tiempo de espera entre que se enciende remotamente un ordenador y llega a la ventana de solicitar el nombre de usuario para realizar el *login*.
- los nombres de los ordenadores que intervienen en el despliegue

```

1 <?xml version="1.0"?>
2 <?xml-stylesheet type='text/xsl' href='style.xsl'?>
3 <Clonar>
4   <!--Parametros del despliegue-->
5   <Parametros>
6     <!--Tiempo de espera en segundos-->
7     <TiempoEsperaLogin>240</TiempoEsperaLogin>
8   </Parametros>
9   <!--Lista de ordenadores para desplegar-->
10  <Ordenadores>
11    <Ordenador><Nombre>ORDENADOR01</Nombre></Ordenador>
12    <Ordenador><Nombre>ORDENADOR02</Nombre></Ordenador>
13    <Ordenador><Nombre>ORDENADOR03</Nombre></Ordenador>
14    <Ordenador><Nombre>ORDENADOR04</Nombre></Ordenador>
15    <Ordenador><Nombre>ORDENADOR05</Nombre></Ordenador>
16    <Ordenador><Nombre>ORDENADOR06</Nombre></Ordenador>
17    <Ordenador><Nombre>ORDENADOR07</Nombre></Ordenador>
18  </Ordenadores>
19 </Clonar>

```

Código 8: Contenido del fichero ORDENADORES_DESPLIEGUE.XML.

En este ejemplo del fichero XML (ver Código 8), el tiempo de espera para que el cliente llegue a solicitar el *login*, era de 240 segundos, e intervenían siete ordenadores en el despliegue.

4.9.6.3. Crear tarea programada en el servidor MDT01

Ahora solo falta crear una tarea programada en el servidor MDT01, que ejecute el *script* de despliegue automático en los ordenadores. Se asume que existen en la carpeta C:\Despliegue, los siguientes ficheros: DespliegueAutomatico.ps1 y ORDENADORES_DESPLIEGUE.xml



Se va a crear la tarea programada y para ello, pulsamos en administración de equipos, programador de tareas, biblioteca del programador de tareas, Crear tarea y se van rellenando los datos solicitados:

Pestaña General

1. Nombre: Despliegue automático
2. Descripción: Ejecuta el *script* DespliegueAutomatico.ps1 para iniciar despliegues automáticos
3. Se selecciona: Ejecutar tanto si el usuario inició sesión como si no
4. Se selecciona: Ejecutar con los privilegios más altos
5. Configurar para: Windows Server 2012 R2

Pestaña Desencadenadores

Se tiene que programar cuando queremos que se ejecute la tarea de despliegue. No nos interesaba que se ejecute durante las horas lectivas, así que se programa para que se ejecute por la noche, a las 22:00 horas. Como es una tarea que tendremos que confirmar cuando nos interesa que se ejecute, tendremos que ir modificándola en función de nuestras necesidades.

Pestaña Acciones

Se seleccionó la opción Nueva:

1. Acción: Iniciar un programa
2. Programa o script: powershell.exe
3. Agregar argumentos (opcional): C:\Despliegue\DespliegueAutomatico.ps1
4. Iniciar en (opcional): C:\Despliegue

4.9.6.4. Ejecutar la tarea en el servidor MDT01

Cuando se lanza la tarea programada en el servidor y se ejecuta el *script*, se crea un fichero de registro en la misma carpeta donde se encuentra ubicado el *script*, con el resultado de la ejecución. En Código 9 se muestra un ejemplo del fichero de registro que genera.

```
1 26/05/2016 09:56:36 =====
2 26/05/2016 09:56:36 Lanzando despliegue desatendido v1.2 25-5-2016
3 26/05/2016 09:56:36 Leyendo el fichero ORDENADORES_DESPLIEGUE.xml
4 26/05/2016 09:56:37 Número de PCs en fichero es 7
5 26/05/2016 09:56:37 Tiempo de espera para el Login es 240 segundos.
6 26/05/2016 09:56:50 Despertando a ORDENADOR01
7 26/05/2016 09:56:52 Despertando a ORDENADOR02
8 26/05/2016 09:57:05 Despertando a ORDENADOR03
9 26/05/2016 09:57:07 Despertando a ORDENADOR04
10 26/05/2016 09:57:09 Despertando a ORDENADOR05
11 26/05/2016 09:57:12 Despertando a ORDENADOR06
12 26/05/2016 09:57:12 Despertando a ORDENADOR07
```

```
13 26/05/2016 09:57:56 7 LOGIN siguen en marcha, LOGIN_ORDENADOR01, LOGIN_
ORDENADOR02, LOGIN_ORDENADOR03, LOGIN_ORDENADOR04, LOGIN_ORDENADOR05,
LOGIN_ORDENADOR06, LOGIN_ORDENADOR07
14 26/05/2016 09:58:01 4 LOGIN siguen en marcha, LOGIN_ORDENADOR02, LOGIN_
ORDENADOR04, LOGIN_ORDENADOR05, LOGIN_ORDENADOR07
15 26/05/2016 09:58:12 2 LOGIN siguen en marcha, LOGIN_ORDENADOR05, LOGIN_
ORDENADOR07
16 26/05/2016 09:59:47 1 LOGIN siguen en marcha, LOGIN_ORDENADOR05
17 26/05/2016 09:59:47 Intentando lanzar la tarea remota DesplegarMDTLab
en ORDENADOR01
18 26/05/2016 09:59:47 CORRECTO: se ha intentado ejecutar la tarea
programada "DesplegarMDTLab".
19 26/05/2016 09:59:47 Intentando lanzar la tarea remota DesplegarMDTLab
en ORDENADOR02
20 26/05/2016 09:59:47 CORRECTO: se ha intentado ejecutar la tarea
programada "DesplegarMDTLab".
21 26/05/2016 09:59:47 Intentando lanzar la tarea remota DesplegarMDTLab
en ORDENADOR03
22 26/05/2016 09:59:47 CORRECTO: se ha intentado ejecutar la tarea
programada "DesplegarMDTLab".
23 26/05/2016 09:59:47 Intentando lanzar la tarea remota DesplegarMDTLab
en ORDENADOR04
24 26/05/2016 09:54:48 CORRECTO: se ha intentado ejecutar la tarea
programada "DesplegarMDTLab".
25 26/05/2016 09:54:48 ERROR: No HAY una sesión de adminMDT abierta en
ORDENADOR05
26 26/05/2016 09:59:48 ERROR: NO SE HA EJECUTADO la tarea programada
DespliegueMDTLab en ORDENADOR05
27 26/05/2016 09:59:48 Intentando lanzar la tarea remota DesplegarMDTLab
en ORDENADOR06
28 26/05/2016 09:59:48 CORRECTO: se ha intentado ejecutar la tarea
programada "DesplegarMDTLab".
29 26/05/2016 09:59:48 Intentando lanzar la tarea remota DesplegarMDTLab
en ORDENADOR07
30 26/05/2016 09:59:48 CORRECTO: se ha intentado ejecutar la tarea
programada "DesplegarMDTLab".
31 26/05/2016 09:59:48 FIN de la ejecución del despliegue
```

Código 9: Contenido del fichero DESPLIEGUE_MDT01.LOG.

Como vemos en las líneas 25 y 26 de Código 9, en el **ORDENADOR05** no se ha lanzado la ejecución de la tarea por algún problema, puede que:

- Haya fallado el encendido remoto por medio de Wake-On-Lan.
- Presente algún problema de conectividad de red.
- No se haya iniciado la sesión de escritorio remoto.

En los ordenadores donde se ha ejecutado la tarea programada se lanzará el despliegue y se hará de forma automática, sin necesitar ninguna intervención por parte del usuario o técnico.



5. Conclusiones

Se ha conseguido el objetivo de presente TFG, que era el seleccionar una herramienta que cumple con nuestras necesidades para hacer despliegues de imágenes personalizadas del sistema operativo dentro de nuestro entorno corporativo.

La herramienta seleccionada ha sido Microsoft Deployment Toolkit (MDT) que, como hemos visto, nos permite:

- Hacer despliegues LTI y ZTI.
- Tener un repositorio de controladores de dispositivos y de aplicaciones.
- Personalizar los despliegues de cada uno de los ordenadores individualmente.

Los motivos para la elección de MDT han sido:

1. Precio. La licencia de uso es gratuita.
2. Se integra con la infraestructura actual de los servidores de la ETSIAMN.
3. Se permiten configuraciones específicas para cada ordenador, tanto de controladores de dispositivos como de aplicaciones.
4. Capacidad de expansión. El entorno de MDT nos permite añadir *scripts* y configuraciones externas al propio programa, para adecuarlo a nuestras necesidades.
5. Facilidad de uso. Una vez configurado correctamente, cualquier técnico puede hacer un despliegue desatendido a petición; utilizando un medio físico o un arranque por PXE.
6. Automatización¹. Aunque en principio MDT solo permite despliegues LTI, como hemos visto, podemos hacer despliegues ZTI sí:
 - Almacenamos información de los ordenadores en una base de datos de SQL.
 - Añadimos configuración adicional a las reglas del DS.
 - Arrancamos los ordenadores en el entorno PXE.

La herramienta MDT lleva en producción en la ETSIAMN desde hace varios meses con resultados muy productivos. Desde su implementación hemos actualizado el 100 % de los ordenadores de las aulas de informática, desde el sistema operativo Windows 7 de 32 bits a la última versión del sistema operativo Windows 10 de 64 bits.

La capacidad de expansión de la aplicación, utilizando *scripts* externos de Windows Powershell, la convierte en una herramienta muy versátil y potente, lo que nos ha permitido realizar despliegues desatendidos y programados fuera del horario lectivo. También nos

¹ Lo cierto es, que como tenemos que pulsar una tecla, para arrancar los ordenadores (normalmente [F12]), no es realmente ZTI, pero es lo único que el usuario o el técnico tiene que hacer para iniciar un despliegue. La ejecución de un *script* de forma remota por medio de una tarea programada en el servidor, consigue realizar despliegues ZTI.



ha permitido el personalizar totalmente las imágenes a nuestras necesidades, como por ejemplo, el instalar algunas aplicaciones específicas a ordenadores concretos.

A nivel personal este TFG y el trabajar con la herramienta MDT, me ha servido para conocer más a fondo como funciona internamente el sistema operativo Windows, como gestiona e implementa los controladores de dispositivos, y como realiza los procesos de captura y despliegue de las imágenes personalizadas.

He programado *scripts* utilizando Windows Powershell, que me han permitido el poder hacer los despliegues totalmente desatendidos y programados. He aprendido a utilizar las clases de WMI del sistema operativo Windows, que gestionan el hardware y el software del ordenador, lo que nos permite recuperar remotamente información de todo tipo de los ordenadores de las aulas de informática, desde conocer el fabricante del ordenador, saber que programas tiene instalados, o que usuarios han iniciado sesión alguna vez en el ordenador.

6. Trabajos futuros

El servicio de informática de la ETSIAMN va a seguir trabajando con la herramienta MDT, y se piensa implementar diversas mejoras como:

Añadir más funcionalidades al *script* de despliegue automático:

- Crear un registro de actividad del despliegue por ordenador.
- Añadir un contador de tiempo para hacer un seguimiento de los tiempos de despliegue desde la parte del ordenador cliente.
- Utilizar comandos específicos de Windows Powershell para recuperar información de los despliegues a medida que se realizan e integrar esos datos en los ficheros de registro.
- Controlar cuando finaliza el despliegue, para poder apagar los ordenadores que intervienen en el despliegue remotamente.

En cuanto a las secuencias de tareas:

- Crear un *script* que guarde información de los tiempos de ejecución desde la parte del servidor, y que funcione en combinación con el *script* de despliegue automático. Por ejemplo: monitorizar justo antes de desplegar la imagen, y al finalizar el despliegue.

Respecto a la base de datos de MDT:

- Crear *scripts* que permitan realizar copias de seguridad de la base datos y recuperarlas.
- Crear *scripts* que permitan recopilar información de múltiples ordenadores utilizando Windows Powershell y WMI, y la añadir esa información masivamente a la base de datos de MDT.
- Alterar la estructura de la base de datos de MDT, para que contenga otra información relevante de los ordenadores.

7. Bibliografía

- [1] WIKIPEDIA. Hardware. [en línea] <https://es.wikipedia.org>. 2016 [consulta: 13 mayo 2016]. Disponible en: <https://es.wikipedia.org/wiki/Hardware>
- [2] WIKIPEDIA. Software. [en línea] <https://es.wikipedia.org>. 2016 [consulta: 13 mayo 2016]. Disponible en: <https://es.wikipedia.org/wiki/Software>
- [3] OPENGNSYS. Restauración y despliegue de imágenes. [en línea] <http://opengnsys.es>. 2015 [consulta: 5 mayo 2016]. Disponible en: <http://opengnsys.es/wiki/DocumentacionUsuario1.0.6/RestauracionDespliegue>
- [4] OPENGNSYS. [en línea] <http://opengnsys.es>. 2016 [consulta: 5 mayo 2016]. Disponible en: <http://opengnsys.es>
- [5] OPENGNSYS. Preparación equipo modelo antes de hacer las imágenes. [en línea] <http://opengnsys.es>. 2014 [consulta: 5 mayo 2016]. Disponible en: <http://www.opengnsys.es/wiki/EquipoModelo>
- [6] MICROSOFT. Información general sobre los Servicios de implementación de Windows. [en línea] <https://technet.microsoft.com>. 2016 [consulta: 6 mayo 2016]. Disponible en: <https://technet.microsoft.com/es-es/library/hh831764.aspx>
- [7] SYMANTEC. Hoja de datos. [en línea] <https://www.symantec.com>. 2016 [consulta: 6 mayo 2016]. Disponible en: <https://www.symantec.com/es/es/ghost-solution-suite>. Ruta de acceso: Descargar hoja de datos. Disponible en PDF en: <https://www.symantec.com/content/dam/symantec/docs/data-sheets/ghost-solutions-suite-en.pdf>.
- [8] MICROSOFT TECHNET. Microsoft Deployment Toolkit [en línea] <https://technet.microsoft.com>. 2016 [consulta: 6 mayo 2016]. Disponible en: <https://technet.microsoft.com/es-es/windows/dn475741>
- [9] MICROSOFT TECHNET. Key Features in MDT 2013 [en línea] <https://technet.microsoft.com>. 2014 [consulta: 6 mayo 2016]. Disponible en: <https://technet.microsoft.com/es-es/library/dn744277.aspx>
- [10] ACRONIS. Hoja de datos. [en línea] <http://www.acronis.com>. 2016 [consulta: 7 mayo 2016]. Disponible en: <http://www.acronis.com/es-es/business/enterprise-solutions/image-deployment/>. Ruta de acceso: Hoja de datos. Disponible en PDF en: <http://www.acronis.com/es-es/download/docs/asd5/datasheet>.
- [11] SEARCHWINDOWSSERVER. Microsoft System Center Configuration Manager 2012 [en línea] <http://searchwindowsserver.techtarget.com>. 2012 [consulta: 7 mayo 2016]. Disponible en: <http://searchwindowsserver.techtarget.com/definition/Microsoft-System-Center-Configuration-Manager-2012>

- [12] MICROSOFT. System Center Configuration Manager [en línea] <https://www.microsoft.com>. 2016 [consulta: 7 mayo 2016]. Disponible en: <https://www.microsoft.com/en-us/server-cloud/products/system-center-configuration-manager/features.aspx>
- [13] MICROSOFT. MDT TechNet Forum - FAQ & Getting Started Guide [en línea] <https://social.technet.microsoft.com>. 2015 [consulta: 7 mayo 2016]. Disponible en: <https://social.technet.microsoft.com/Forums/en-US/ef0a9119-63b6-45ae-a871-866ea62ee3e8/mdt-technet-forum-faq-getting-started-guide?forum=mdt>
- [14] MICROSOFT. Microsoft Deployment Toolkit [en línea] <https://technet.microsoft.com>. 2016 [consulta: 7 mayo 2016]. Disponible en: <https://technet.microsoft.com/es-es/windows/dn475741.aspx>
- [15] MICROSOFT. Using the Microsoft Deployment Toolkit, , Choosing LTI, ZTI, or UDI Deployments [en línea] <https://technet.microsoft.com>. 2016 [consulta: 7 mayo 2016]. Disponible en: <https://technet.microsoft.com/en-us/library/dn759415.aspx>
- [16] MICROSOFT. Usar SQL Server Management Studio [en línea] <https://msdn.microsoft.com>. 2016 [consulta: 5 abril 2016]. Disponible en: <https://msdn.microsoft.com/es-es/library/ms174173.aspx>
- [17] MICROSOFT. MDT 2013 Lite Touch Components [en línea] <https://technet.microsoft.com>. 2014 [consulta: 5 abril 2016]. Disponible en: <https://technet.microsoft.com/en-us/library/dn744285%28v=ws.11%29.aspx>
- [18] MICROSOFT. How to create a hidden share in a network [en línea] <https://answers.microsoft.com>. 2010 [consulta: 30 mayo 2016]. Disponible en: http://answers.microsoft.com/en-us/windows/forum/windows_xp-networking/how-to-create-a-hidden-share-in-a-network/1d2c75a3-5feb-48d4-9dbe-e2dba34ef6fd?auth=1
- [19] MICROSOFT. *Hyper-V Feature Overview* [en línea] <https://msdn.microsoft.com>. 2016 [consulta: 5 abril 2016]. Disponible en: <https://msdn.microsoft.com/en-us/library/cc768521%28v=bts.10%29.aspx>
- [20] MICROSOFT. Personalizar Windows en modo auditoría [en línea] <https://technet.microsoft.com>. 2016 [consulta: 22 abril 2016]. Disponible en: <https://technet.microsoft.com/es-es/library/cc722413%28v=ws.10%29.aspx>
- [21] MICROSOFT. Funcionamiento de Sysprep [en línea] <https://technet.microsoft.com>. 2016 [consulta: 20 mayo 2016]. Disponible en: <https://technet.microsoft.com/es-es/library/cc766514%28v=ws.10%29.aspx>
- [22] KASPERSKY. How to install and remove Kaspersky Endpoint Security 10 for Windows via the command prompt. Silent installation. [en línea] <https://support.kaspersky.com>. 2016 [consulta: 23 mayo 2016]. Disponible en: <https://support.kaspersky.com/9363#block0>



[23] DEPLOYMENTRESEARCH. MDT 2013 Lite Touch Driver Management [en línea] <http://deploymentresearch.com>. 2013 [consulta: 10 noviembre 2015]. Disponible en: <http://deploymentresearch.com/Research/Post/325/MDT-2013-Lite-Touch-Driver-Management>

[24] MICROSOFT. *Troubleshooting and Tips* [en línea] <https://technet.microsoft.com>. 2004 [consulta: 4 febrero 2016]. Disponible en: <https://technet.microsoft.com/es-es/library/ee692772.aspx>

[25] MICROSOFT. *Scripting con Windows PowerShell* [en línea] <https://technet.microsoft.com>. 2014 [consulta: 7 marzo 2016]. Disponible en: <https://technet.microsoft.com/es-es/library/bb978526.aspx>

[26] MICROSOFT. *Win32_ComputerSystemProduct class* [en línea] <https://msdn.microsoft.com>. 2016 [consulta: 16 enero 2016]. Disponible en: <https://msdn.microsoft.com/en-us/library/aa394105%28v=vs.85%29.aspx>

[27] MICROSOFT. *Win32_NetworkAdapterConfiguration class* [en línea] <https://msdn.microsoft.com>. 2016 [consulta: 16 enero 2016]. Disponible en: <https://msdn.microsoft.com/en-us/library/aa394217%28v=vs.85%29.aspx>

[28] MICROSOFT. *Win32_BaseBoard class* [en línea] <https://msdn.microsoft.com>. 2016 [consulta: 16 enero 2016]. Disponible en: <https://msdn.microsoft.com/en-us/library/aa394072%28v=vs.85%29.aspx>

[29] MICROSOFT. *Use the MDT database to stage Windows 10 deployment information* [en línea] <https://technet.microsoft.com>. 2016 [consulta: 16 mayo 2016]. Disponible en: <https://technet.microsoft.com/en-us/itpro/windows/deploy/use-the-mdt-database-to-stage-windows-10-deployment-information>

Abreviaturas y siglas

ACT: Kit de compatibilidad de aplicaciones

AD: Active Directory

ADK: Assessment and Deployment Kit

AS&RS: Analysis Services & Reporting Services

BIOS: Basic Input Output System

BD: Base de datos

CD: disco óptico que se graba en forma digital, lo que permite acumular una gran cantidad de información.

CPU: Central Processing Unit

DaRT: Microsoft Diagnostics and Recovery Toolset

DFSR: Distributed File System Replication

DHCP: Dynamic Host Configuration Protocol

DISM: Deployment Image Servicing and Management

DNS: Domain Name System

DS: Deployment Share

DVD: Digital Versatile Disc o Digital Video Disc. Disco óptico capaz de contener en forma codificada gran cantidad de imágenes, sonidos o datos.

DW: Deployment Workbench

GPO: Group Policy Object

GPT: GUID Partition Table

GUID: Globally Unique Identifier

Hyper-V: Plataforma de virtualización de Microsoft.

ICD: Diseñador de imágenes y configuraciones

ISO (archivo): Una imagen ISO es un archivo informático donde se almacena una copia o imagen exacta de un sistema de archivos o ficheros de un disco óptico, normalmente un disco compacto (CD) o un disco versátil digital (DVD). Se rige por el estándar ISO 9660, que le da nombre.

KMS: Key Management Service

LDS: Lightweight Directory Service

LTI: Lite Touch Installation

MAK: Multiple Activation Key

MDM: Mobile Device Management

MDT: Microsoft Deployment Toolkit

MSI: Microsoft Installer

OEM: Original Equipment Manufacturer

OOBE: Out-Of-Box Experience

OS: Operating System

OSD: Operating System Deployment

OU: Organizational Unit

PNP: Plug and Play

PNP ID: Plug and Play IDentification

PXE: Preboot eXecution Environment

RAM: Random Access Memory

RDP: Remote Desktop Protocol

RPC: Remote Procedure Call

SCCM: System Center Control Manager

SKU: Stock Keeping Unit

SQL: Structured Query Language

SSMS: SQL Server Management Studio

TCP/IP: Transmission Control Protocol/Internet Protocol

TI: Tecnologías de la información

TIC: Tecnologías de la información y comunicación

TS: Task Sequence

UAC: User Access Control

USB: Universal Serial Bus

UDI: User Driven Installation

UEFI: Unified Extensible Firmware Interface

UI: User Interface

UFD: USB Flash Drive

URL: Uniform Resource Locator

USMT: User State Migration Tool

UUID: Universal Unique IDentifier
VAMT: Volume Activation Management Tool
VBScript: Visual Basic Script
VLAN: Virtual Local Area Network
VHD: Virtual Hard Disk
WAIK: Windows Assessment and Deployment Kit
WDS: Windows Deployment Services
WEB: World Wide Web
WIM: Windows Imaging File Format
WinPE: Windows Preinstallation Environment
WMI: Windows Management Instrumentation
WOL: Wake-On-LAN
WSIM: Windows System Image Manager
WSUS: Windows Server Update Services
ZTI: Zero Touch Installation



Anexos

Anexo 1. El BUG de Windows 10

Las primeras pruebas de despliegue con MDT las hice con el sistema operativo Windows 8.1, donde todo funcionaba correctamente; particularmente la inyección de controladores de dispositivos en función de la detección del fabricante y el modelo del ordenador. La estructura de controladores de dispositivos que tenía era la siguiente:

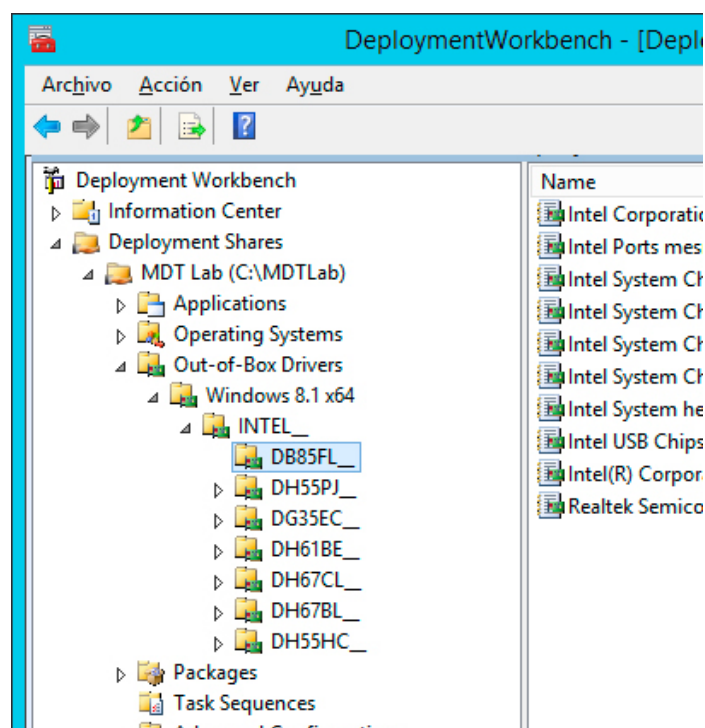


Figura A1: Estructura de carpetas de controladores para Windows 8.1 en el DS.

Para la inyección correcta de controladores hay que asignar la variable DriverGroup001 a Windows 8.1 x64\%Manufacturer%\%Model% creando un nuevo paso en la secuencia de tareas, Set Task Sequence Variable, justo antes del paso que inyecta los controladores de dispositivo, y realizar la asignación:

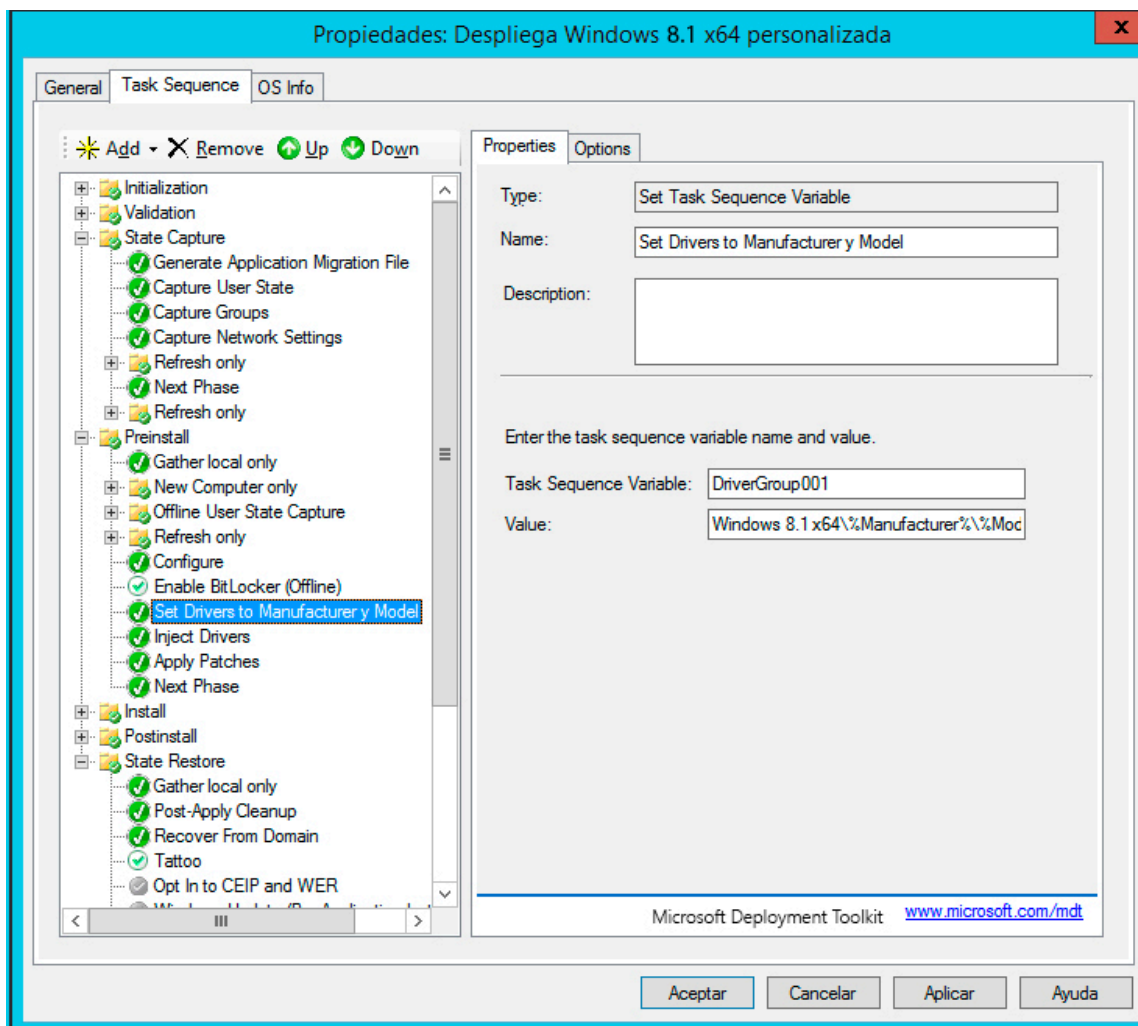


Figura A2: Asignación de la variable DriverGroup001 a Windows 8.1 x64\%Manufacturer%\%Model%.

En ese momento absolutamente todas las placas base de los ordenadores que teníamos estaban fabricadas por Intel, y teníamos siete modelos diferentes, pues se fueron comprando a lo largo del tiempo. Cuando Microsoft publicó el sistema operativo Windows 10 y estuvieron disponibles las herramientas de despliegue para Windows 10, tuve que crear una estructura de carpetas nueva para los controladores de dispositivos de Windows 10.

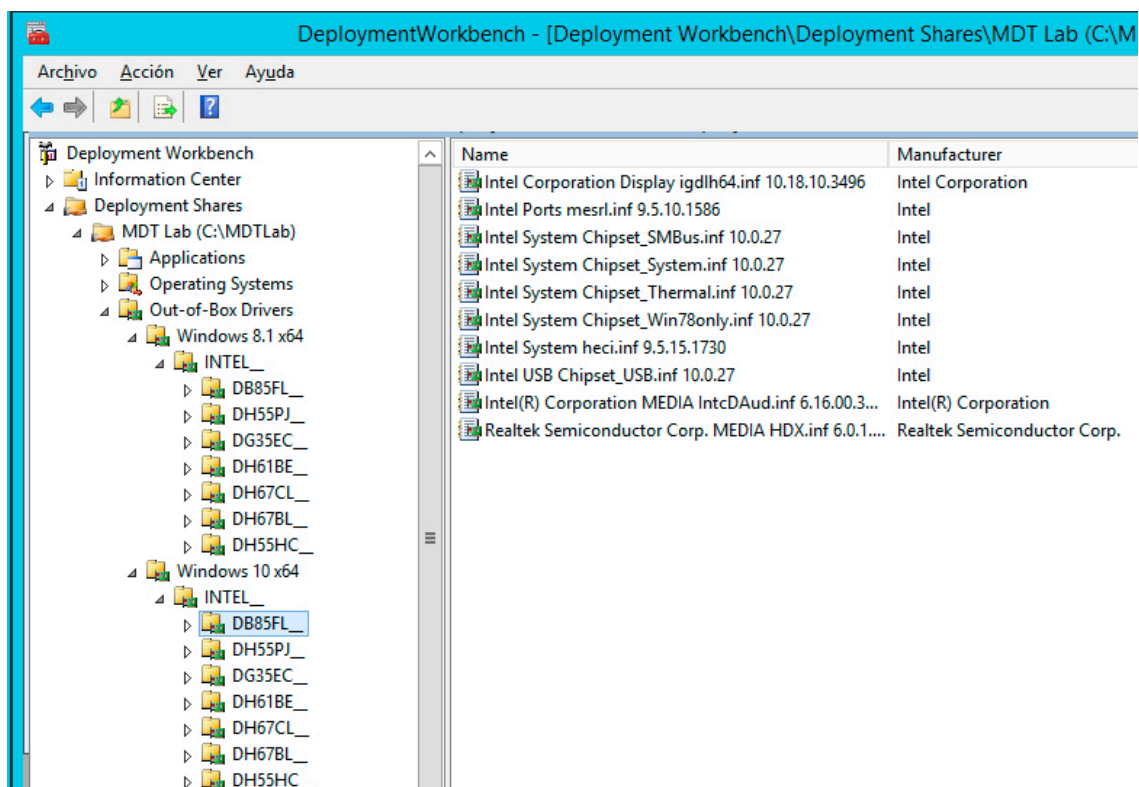


Figura A3: Estructura de carpetas de controladores para Windows 8.1 y para Windows 10 en el DS.

En las primeras pruebas que realicé con MDT, observé que, cuando MDT ejecutaba los comandos de inyectar los controladores de dispositivos no funcionaba correctamente, no inyectaba ningún controlador. Era como si no fuera capaz de determinar el fabricante y el modelo del ordenador.

Después de investigar y determinar que el *script* ZTigahter.wsf de MDT, es el encargado de recopilar información sobre el ordenador donde se realiza el despliegue, y hacer una simulación sobre el ordenador de la siguiente orden de Windows Powershell:

```
Get-WmiObject -Class Win32_ComputerSystemProduct
```

Devuelve valores nulos, o en blanco de los datos del fabricante y del modelo del ordenador.

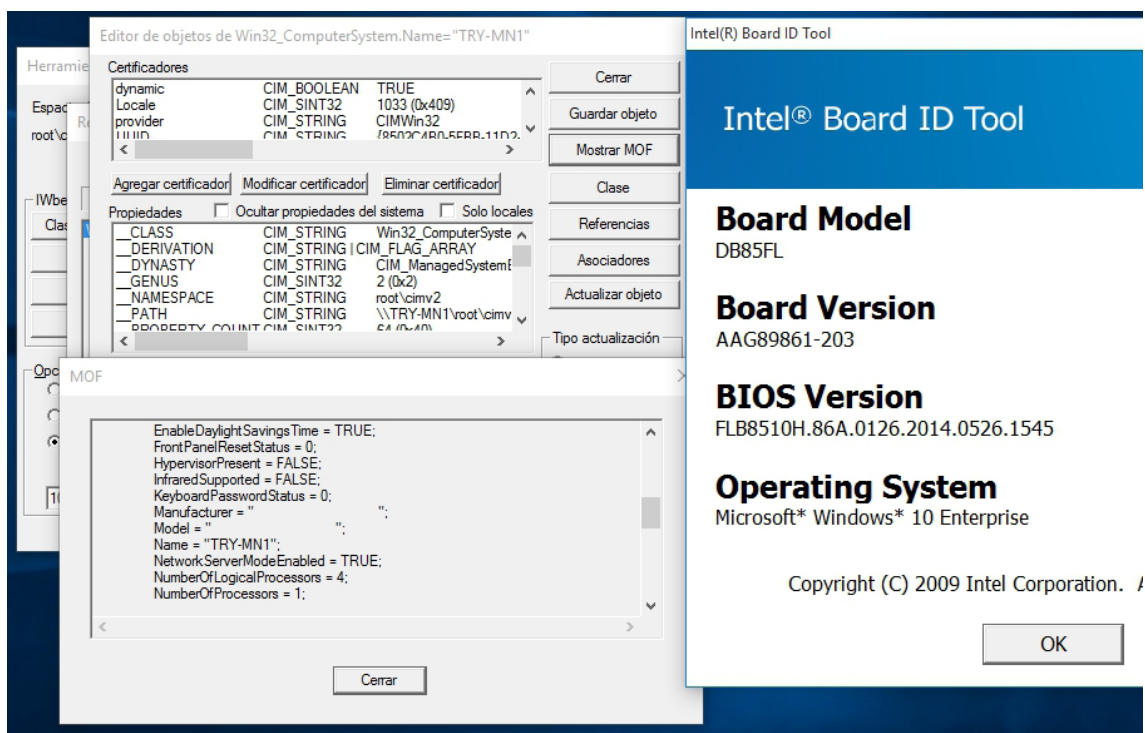


Figura A4: Captura de pantalla de Windows 10 con los datos del fabricante y del modelo en blanco.

Mientras que la ejecución del comando en el mismo ordenador cuando ejecuta Windows 8.1 es la siguiente:

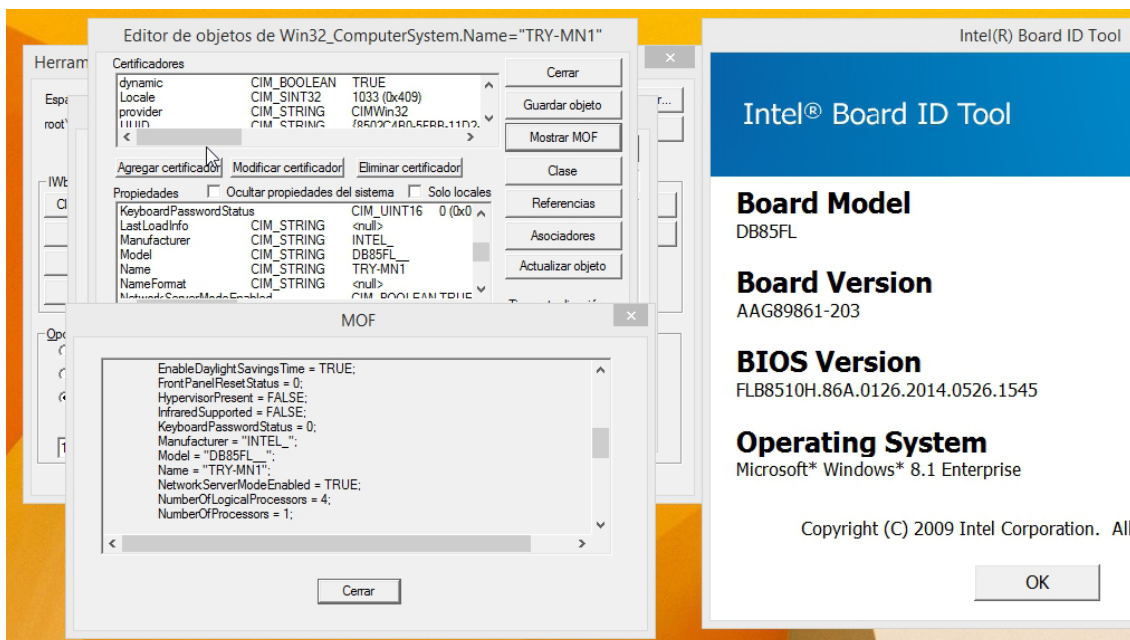


Figura A5: Captura de pantalla de Windows 8.1 con los datos del fabricante y del modelo.

Dado que no encontré ninguna solución utilizando el buscador de Google, me decidí

a abrir un hilo en los foros de soporte técnico de Microsoft, que se puede consultar en la siguiente URL:

<https://social.technet.microsoft.com/Forums/en-US/18bba33c-490d-4334-a88f-af18c4c791e6/mdt-2013-update-1-not-injecting-drivers?forum=mdt>

The screenshot shows a forum post on Microsoft Technet. The title is "MDT 2013 Update 1 not injecting drivers". The author is Daniel Jelinek nlx, a Partner, who joined in June 2011. The post is dated Tuesday, September 01, 2015, at 11:07 AM. The user reports a problem where MDT 2013 Update 1 fails to inject drivers for Windows 10 after an upgrade from Windows 8.1. The user provides a PowerShell command and its output, and asks for a workaround. The thread has 5 points and 0 votes.

Figura A6: Hilo creado en el foro de MDT en Microsoft Technet.

Obtuve varias respuestas al problema, pero ninguna solución. Finalmente, Keith Garner, uno de los moderadores del foro concluyo que el fallo no estaba relacionado con el software MDT, sino que era un fallo relativo al propio sistema operativo Windows 10, y al Windows PE de Windows 10, y como una vía de continuar y de encontrar una solución, me propuso el abrir un caso con Microsoft. Tarea que intente, pero para mi sorpresa, no es tan fácil contactar con una corporación como Microsoft para darles a conocer la existencia del fallo.

Como el problema está relacionado con las placas base fabricadas por Intel, me puse

en contacto con el proveedor que nos suministró las placas base, y accedió a contactar por mí con los ingenieros de Intel en Irlanda. Tras exponer el caso, después de varios días, me remitieron a la siguiente página WEB de Johan Arwidmark, indicándome que utilizara el segundo escenario posible para la inyección de los controladores de dispositivos:

<http://deploymentresearch.com/Research/Post/325/MDT-2013-Lite-Touch-Driver-Management>

El segundo escenario está basado en añadir predictibilidad cuando se inyectan los controladores, creando un perfil de selección que le indica a la secuencia de tareas que controladores debe de inyectar en el despliegue. El inconveniente que tiene este escenario, es que hay que crear tantas secuencias de tareas como distintos modelos de ordenador tengamos. La solución evidentemente funciona, pero no me acababa de gustar.

Después de un mes publicar el hilo, finalmente otra persona, Daniel Jelinek, escribió que tenía exactamente el mismo problema, pero lo resolvió realizando la siguiente *query*:

```
SELECT * From Win32_BaseBoard where product like "DZ77SL%"
```

en la secuencia de tareas, de modo que, si devuelve verdadero es que es el ordenador que tiene una placa fabricada por Intel y de ese modelo concreto. En mi caso tengo siete modelos distintos de placas base y no era viable esa solución. Como ya había averiguado que el *script* que realiza la recopilación de datos del ordenador es ZTIGahter.wsf, me propuse modificarlo para adecuarlo a mis necesidades. El problema es que no es una solución permanente y requiere alterar los *scripts* originales de Microsoft.

Este es el fragmento del *script* modificado.

```
1  ' Get the product from the Win32_BaseBoard class
2  Set objResults = objWMI.InstancesOf("Win32_BaseBoard")
3
4  For each objInstance in objResults
5      If not IsNull(objInstance.Product) then
6          sProduct = Trim(objInstance.Product)
7      End if
8  Next
9
10 If sProduct = "" then
11     oLogging.CreateEntry "Unable to determine product via WMI.",
LogTypeInfo
12 End if
13
14 ' Get the make, model, and memory from the Win32_ComputerSystem class
15
16 Set objResults = objWMI.InstancesOf("Win32_ComputerSystem")
17
18 For each objInstance in objResults
19     If not IsNull(objInstance.Manufacturer) then
20         sMake = Trim(objInstance.Manufacturer)
```



```
21 ' si sMake es una cadena vacia, significa que la placa es Intel,
22 ' entonces asigno manualmente el INTEL_ a la variable sMake
23 If sMake= "" then
24     sMake = "INTEL_"
25 End if
26
27 End if
28 If not IsNull(objInstance.Model) then
29     sModel = Trim(objInstance.Model)
30
31 ' si la variable sMake es igual a INTEL_, cambio el valor de la
32 ' variable sModel
33 If sMake = "INTEL_" then
34     sModel = sProduct & "_"
35 End If
36 End if
37 If not IsNull(objInstance.TotalPhysicalMemory) then
38     sMemory = Trim(Int(objInstance.TotalPhysicalMemory / 1024 / 1024))
39 End if
40 Next
```

Código A1. Líneas modificadas del fichero ZTIGather.WSF.

De este modo podía seguir manteniendo la estructura de carpetas para los controladores y todo volvía a funcionar correctamente.

Finalmente, encontré una solución más efectiva, permanente, y que no requiere modificar los *scripts* originales de MDT. En la página web de Mikael Nystrom, en la siguiente URL:

<https://deploymentbunny.com/2015/11/24/osd-deployment-deploying-intel-nuc-and-getting-drivers-and-settings-assigned-using-the-aliasuserexit-vbs-converting-product-into-modelaliases/>

La solución fue: aprovechar la capacidad de expansión de la configuración de MDT con *scripts* externos, y añadir un *script* que permita recuperar toda la información posible del ordenador. En función de la información recopilada, se generará una variable que contendrá un nombre más amigable para la combinación de fabricante y modelo del ordenador, y que se utilizará posteriormente cuando se realice el proceso de inyección de los controladores de dispositivos. El *script* se llama AliasUserExit.vbs, y hay que modificarlo para adecuarlo a nuestras necesidades. Además, debemos de copiar el *script* AliasUserExit.vbs en la carpeta Scripts del DS MDTLab para que funcione.

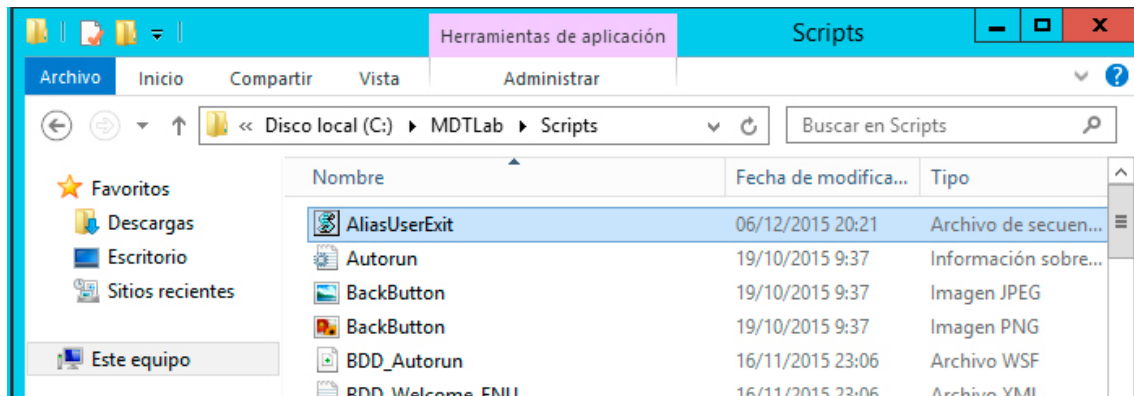


Figura A7: Carpeta Scripts con el fichero AliasUserExit.vbs.

```

1  '---
2  ' Solution: Sample Files
3  ' Purpose: Custom Script for assign friendly Make and Model alias and
4  ' to extend hardware inventory
5  ' Version: 1.5 - Okt 10, 2015 - Johan Arwidmark & Mikael Nystrom
6  '
7  ' This script is based of Microsoft Sample Code from the deployment
8  ' guys blog
9  ' (http://blogs.technet.com/b/deploymentguys) and as such we need to
10 ' have a
11 ' copyright statement. Special thanks goes to Ben Hunter, Michael
12 ' Murgolo and Steven Markegene.
13 '
14 '
15 ' COPYRIGHT STATEMENT
16 ' This script is provided "AS IS" with no warranties, confers no
17 ' rights and
18 ' is not supported by the authors or Deployment Artist.
19 '---
20 '
21 ' Usage: Modify CustomSettings.ini similar to this:
22 ' [Settings]
23 ' Priority=SetAlias, Default
24 ' Properties=ModelAlias,MakeAlias,MacAlias,SMBIOSBIOSVersion
25 '
26 ' [SetAlias]
27 ' UserExit=AliasUserExit.vbs
28 ' MakeAlias=#SetMakeAlias()#
29 ' ModelAlias=#SetModelAlias()#
30 ' SMBIOSBIOSVersion=#SetSMBIOSBIOSVersion()#
31 ' MacAlias=#SetMacAlias()#
32 ' ***** End Header *****
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912

```

```

36
37  oLogging.CreateEntry "UserExit: Running function SetMakeAlias ",
LogTypeInfo
38  sMake = oEnvironment.Item("Make")
39  SetMakeAlias = ""
40  oLogging.CreateEntry "UserExit: Make is now " & sMake, LogTypeInfo
41
42  Select Case sMake
43
44      Case "Dell Computer Corporation", "Dell Inc.", "Dell Computer
Corp."
45          SetMakeAlias = "Dell"
46
47      Case "Matsushita Electric Industrial Co.,Ltd."
48          SetMakeAlias = "Panasonic"
49
50      Case "VMware, Inc."
51          SetMakeAlias = "VMware"
52
53      Case "SAMSUNG ELECTRONICS CO., LTD."
54          SetMakeAlias = "Samsung"
55
56      Case "Microsoft Corporation"
57          SetMakeAlias = "Microsoft"
58
59      ' Si el fabricante es vacio, se trata de una placa fabricad por
Intel
60      Case ""
61          SetMakeAlias = "Intel"
62
63      ' Si el fabricante es System manufacturer, se trata de una placa
fabricad por ASUS
64      Case "System manufacturer"
65          SetMakeAlias = "Asus"
66
67
68      Case Else
69          SetMakeAlias = sMake
70          oLogging.CreateEntry "UserExit: Alias rule not found. MakeAlias
will be set to Make value.", LogTypeInfo
71      End Select
72  oLogging.CreateEntry "UserExit: MakeAlias has been set to " &
SetMakeAlias, LogTypeInfo
73  oLogging.CreateEntry "UserExit: Departing...", LogTypeInfo
74
75  End Function
76
77  Function SetModelAlias()
78
79  oLogging.CreateEntry "UserExit: Running function SetModelAlias",
LogTypeInfo
80  sMake = oEnvironment.Item("Make")
81  sModel = oEnvironment.Item("Model")
82  sProduct = oEnvironment.Item("Product")
83  ' sManufacturer = oEnvironment.Item("Manufacturer")
84  SetModelAlias = ""
85  sCSPVersion = ""
86  sBIOSVersion = ""

```



```
87  sBIOSName = ""
88  oLogging.CreateEntry "UserExit: Make is now " & sMake, LogTypeInfo
89  oLogging.CreateEntry "UserExit: Model is now " & sModel, LogTypeInfo
90  oLogging.CreateEntry "UserExit: Product is now " & sProduct,
LogTypeInfo
91  ` oLogging.CreateEntry "UserExit: Manufacturer is now " &
sManufacturer, LogTypeInfo
92
93
94  Set colComputerSystemProduct = objWMI.ExecQuery("SELECT * FROM Win32_
ComputerSystemProduct")
95  If Err then
96    oLogging.CreateEntry "Error querying Win32_ComputerSystemProduct: "
& Err.Description & " (" & Err.Number & ")", LogTypeError
97  Else
98    For Each objComputerSystemProduct in colComputerSystemProduct
99      If not IsNull(objComputerSystemProduct.Version) then
100        sCSPVersion = Trim(objComputerSystemProduct.Version)
101        oLogging.CreateEntry "UserExit: Win32_ComputerSystemProduct
Version: " & sCSPVersion, LogTypeInfo
102      End If
103
104    Next
105  End if
106  Set colBIOS = objWMI.ExecQuery("SELECT * FROM Win32_BIOS")
107  If Err then
108    oLogging.CreateEntry "Error querying Win32_BIOS: " & Err.
Description & " (" & Err.Number & ")", LogTypeError
109  Else
110    For Each objBIOS in colBIOS
111      If not IsNull(objBIOS.Version) then
112        sBIOSVersion = Trim(objBIOS.Version)
113        oLogging.CreateEntry "UserExit: Win32_BIOS Version: " &
sBIOSVersion, LogTypeInfo
114      End If
115    Next
116  End if
117
118
119  ` Check by Make
120
121  Select Case sMake
122
123    Case "Dell Computer Corporation", "Dell Inc.", "Dell Computer
Corp."
124      ` Use Model with spaces removed
125      ` SetModelAlias = Replace(sModel, " ", "")
126      SetModelAlias = sModel
127
128    Case "Hewlett-Packard"
129      ` Use Model with spaces removed
130      ` SetModelAlias = Replace(sModel, " ", "")
131
132    Select Case sModel
133      Case "HP Compaq nw8240 (PY442EA#AK8)"
134        SetModelAlias = "HP Compaq nw8240"
135      Case "HP EliteBook 8560w (LG663EA#AK8)"
136        SetModelAlias = "HP EliteBook 8560w"
```

```
137         Case "HP EliteBook 8560w Hewlett Packard"
138             SetModelAlias = "HP EliteBook 8560w"
139             ' añadido el 6-12-2015
140         Case "HP Compaq dc7900 Small Form Factor"
141             SetModelAlias = "HP dc7900"
142
143     Case Else
144         SetModelAlias = sModel
145         oLogging.CreateEntry "UserExit: Alias rule not found. ModelAlias
set to Model value." , LogTypeInfo
146     End Select
147
148
149     Case "HP"
150         ' Use Model with spaces removed
151         ' SetModelAlias = Replace(sModel, " ", "")
152         SetModelAlias = sModel
153
154     Case "IBM"
155         ' Use Model with spaces removed
156         ' SetModelAlias = Replace(sModel, " ", "")
157     Select Case sModel
158         Case "---[HS22]---"
159             SetModelAlias = "IBMHS22"
160         Case Else
161             SetModelAlias = sModel
162             oLogging.CreateEntry "UserExit: Alias rule not found. ModelAlias
set to Model value." , LogTypeInfo
163     End Select
164
165     Case "LENOVO"
166         ' Check by Version property of the Win32_ComputerSystemProduct WMI
class first
167         If Not sCSPVersion = "" Then
168             Select Case sCSPVersion
169                 Case "ThinkPad T61p"
170                     SetModelAlias = "ThinkPad T61"
171                 Case Else
172                     ' Use Version with spaces removed
173                     ' SetModelAlias = Replace(sCSPVersion, " ", "")
174                     SetModelAlias = sModel
175             End Select
176         End If
177         ' Check by first 4 characters of the Model
178     If SetModelAlias = "" Then
179         sModelSubString = Left(sModel,4)
180         Select Case sModelSubString
181             Case "1706"
182                 SetModelAlias = "ThinkPad X60"
183             Case Else
184                 SetModelAlias = sModel
185             oLogging.CreateEntry "UserExit: Alias rule not found.
ModelAlias set to Model value.",LogTypeInfo
186         End Select
187     End If
188
189     Case "Matsushita Electric Industrial Co.,Ltd."
190         'Panasonic Toughbook models
```

```
191     If Left(sModel,2) = "CF" Then
192         SetModelAlias = Left(sModel,5)
193     Else
194         SetModelAlias = sModel
195         oLogging.CreateEntry "UserExit: Alias rule not found. ModelAlias
set to Model value.",LogTypeInfo
196     End If
197
198     Case "Microsoft Corporation"
199
200         Select Case sBIOSVersion
201             Case "VRTUAL - 1000831"
202
203                 SetModelAlias = "Hyper-V2008BetaorRC0"
204             Case "VRTUAL - 5000805", "BIOS Date: 05/05/08 20:35:56 Ver:
08.00.02"
205                 SetModelAlias = "Hyper-V2008RTM"
206             Case "VRTUAL - 3000919"
207                 SetModelAlias = "Hyper-V2008R2"
208             Case "VRTUAL - 9001114"
209                 SetModelAlias = "Hyper-V2012BETA"
210             Case "A M I - 2000622"
211                 SetModelAlias = "VS2005R2SP1orVPC2007"
212             Case "A M I - 9000520"
213                 SetModelAlias = "VS2005R2"
214             Case "A M I - 9000816", "A M I - 6000901"
215                 SetModelAlias = "WindowsVirtualPC"
216             Case "A M I - 8000314"
217                 SetModelAlias = "VS2005orVPC2004"
218             Case Else
219                 SetModelAlias = sModel
220                 oLogging.CreateEntry "UserExit: Alias rule not found.
ModelAlias set to Model value.",LogTypeInfo
221         End Select
222
223     Case "Xen"
224         Select Case sCSPVersion
225             Case "4.1.2"
226                 SetModelAlias = "XenServer602"
227             Case Else
228                 SetModelAlias = "XenServer"
229                 oLogging.CreateEntry "UserExit: Alias rule not found.
ModelAlias set to Model value.",LogTypeInfo
230         End Select
231
232     Case "VMware, Inc."
233         SetModelAlias = "VMware"
234
235     Case "Intel"
236         Select Case sProduct
237             Case "GKPPT10H"
238                 SetModelAlias = "Intel NUC DC3217IYE"238
239             Case Else
240                 SetModelAlias = sProduct
241                 oLogging.CreateEntry "UserExit: Alias rule not found.
ModelAlias set to Product." , LogTypeInfo
242         End Select
243
244     ' si no se encuentra la marca, llamamos a Win32_Baseboard y
```

```
averiguamos el fabricante
245     Case Else
246
247     Set colBaseBoard = objWMI.ExecQuery("SELECT * FROM Win32_
BaseBoard")
248     If Err then
249         oLogging.CreateEntry "Error querying Win32_BaseBoard: " & Err.
Description & " (" & Err.Number & ")",
LogTypeError
250     Else
251     For Each objBaseBoard in colBaseBoard
252         If not IsNull(objBaseBoard.Manufacturer) then
253             sManufacturer = Ucase(Trim(objBaseBoard.Manufacturer))
254             oLogging.CreateEntry "UserExit: Win32_BaseBoard
Manufacturer: " & sManufacturer, LogTypeInfo
255         End If
256     Next
257     End if
258
259     Select Case sManufacturer
260
261     ' Si el fabricante es INTEL CORPORATION le añadimos el modelo de
la placa base
262     Case "INTEL CORPORATION"
263         Select Case sProduct
264             Case "DB85FL"
265                 SetModelAlias = "Intel DB85FL"
266
267             Case "DH55PJ"
268                 SetModelAlias = "Intel DH55PJ"
269
270             Case "DG35EC"
271                 SetModelAlias = "Intel DG35EC"
272
273             Case "DH55HC"
274                 SetModelAlias = "Intel DH55HC"
275
276             Case "DH67BL"
277                 SetModelAlias = "Intel DH67BL"
278
279             Case "DH67CL"
280                 SetModelAlias = "Intel DH67CL"
281
282             Case "DH61BE"
283                 SetModelAlias = "Intel DH61BE"
284
285             Case Else
286                 SetModelAlias = sProduct
287                 oLogging.CreateEntry "UserExit: Alias rule for Intel not
found. ModelAlias set to Product." , LogTypeInfo
288
289             ' Fin del select sProduct dentro de Intel
290         End Select
291
292     Case "ASUSTEK COMPUTER INC."
293         Select Case sProduct
294             Case "H110M-K D3"
295                 SetModelAlias = "Asus H110M-K D3"
```

```

296
297         Case Else
298             SetModelAlias = sProduct
299             oLogging.CreateEntry "UserExit: Alias rule for Asus not
found. ModelAlias set to Product." , LogTypeInfo
300
301         'Fin del select sProduct dentro de Asustek
302         End Select
303
304         ' no se encuentra el fabricante de ninguna manera
305         Case Else
306             SetModelAlias = sModel
307             oLogging.CreateEntry "UserExit: Alias rule not found.
ModelAlias set to Model value." , LogTypeInfo
308         ' Fin de Select Case sManufacturer
309         End Select
310
311     ' Fin de Select Case sMake
312     End Select
313
314     oLogging.CreateEntry "UserExit: ModelAlias has been set to " &
SetModelAlias, LogTypeInfo
315     oLogging.CreateEntry "UserExit: Departing...", LogTypeInfo
316
317 End Function
318
319 Function SetSMBIOSBIOSVersion()
320     oLogging.CreateEntry "UserExit: Running function
SetSMBIOSBIOSVersion", LogTypeInfo
321     Dim objWMI
322     Dim objResults
323     Dim objInstance
324     Dim SMBIOSBIOSVersion
325
326     Set objWMI = GetObject("winmgmts:")
327     Set objResults = objWMI.ExecQuery("SELECT * FROM Win32_BIOS")
328     If Err then
329         oLogging.CreateEntry "Error querying Win32_BIOS: " & Err.
Description & " (" & Err.Number & ")", LogTypeEr
ror
330     Else
331         For each objInstance in objResults
332             If Not IsNull(objInstance.SMBIOSBIOSVersion) Then
333                 SMBIOSBIOSVersion = Trim(objInstance.SMBIOSBIOSVersion)
334             End If
335         Next
336     End If
337     SMBIOSBIOSVersion = Replace(SMBIOSBIOSVersion, " ", "")
338     SMBIOSBIOSVersion = Replace(SMBIOSBIOSVersion, ".", "")
339     SetSMBIOSBIOSVersion = SMBIOSBIOSVersion
340     oLogging.CreateEntry "UserExit: SMBIOSBIOSVersion has been set to "
& SMBIOSBIOSVersion, LogTypeInfo
341     oLogging.CreateEntry "UserExit: Departing...", LogTypeInfo
342 End Function
343
344 Function SetMacAlias()
345     oLogging.CreateEntry "UserExit: Running function SetMacAlias ",
LogTypeInfo

```



```

346 sMac = oEnvironment.Item("MacAddress001")
347 SetMacAlias = ""
348 oLogging.CreateEntry "UserExit: MacAddress001 is now " & sMac,
LogTypeInfo
349 SetMacAlias = Replace(sMac, ":", "")
350 oLogging.CreateEntry "UserExit: SetMacAlias has been set to " &
sMac, LogTypeInfo
351 oLogging.CreateEntry "UserExit: Departing...", LogTypeInfo
352 End Function

```

Código A2: Contenido del fichero AliasUserExit.vbs.

Ahora tenemos que actualizar las Rules del DS MDTLab, editando el fichero CustomSettings.ini, para que utilice el *script* AliasUserExit.vbs y se creen las nuevas propiedades dentro de la secuencia de tareas. En Código A3 se observan las modificaciones realizadas.

```

1 [Settings]
2 Priority=SetAlias, CApps, CSettings, Default
3 Properties=MakeAlias, ModelAlias, MacAlias, SMBIOSBIOSVersion
4
5 [SetAlias]
6 UserExit=AliasUserExit.vbs
7 MakeAlias=#SetMakeAlias()#
8 ModelAlias=#SetModelAlias()#
9 SMBIOSBIOSVersion=#SetSMBIOSBIOSVersion()#
10 MacAlias=#SetMacAlias()#

```

Código A3: Líneas modificadas del fichero CUSTOMSETTINGS.INI para utilizar el *script* AliasUserExit.

Después de ejecutar el *script* AliasUserExit.vbs, se le asigna al ordenador un nombre amigable, en la variable %ModelAlias%, de modo que la estructura actual de carpetas de los controladores de dispositivos se puede ver en la Figura A8.

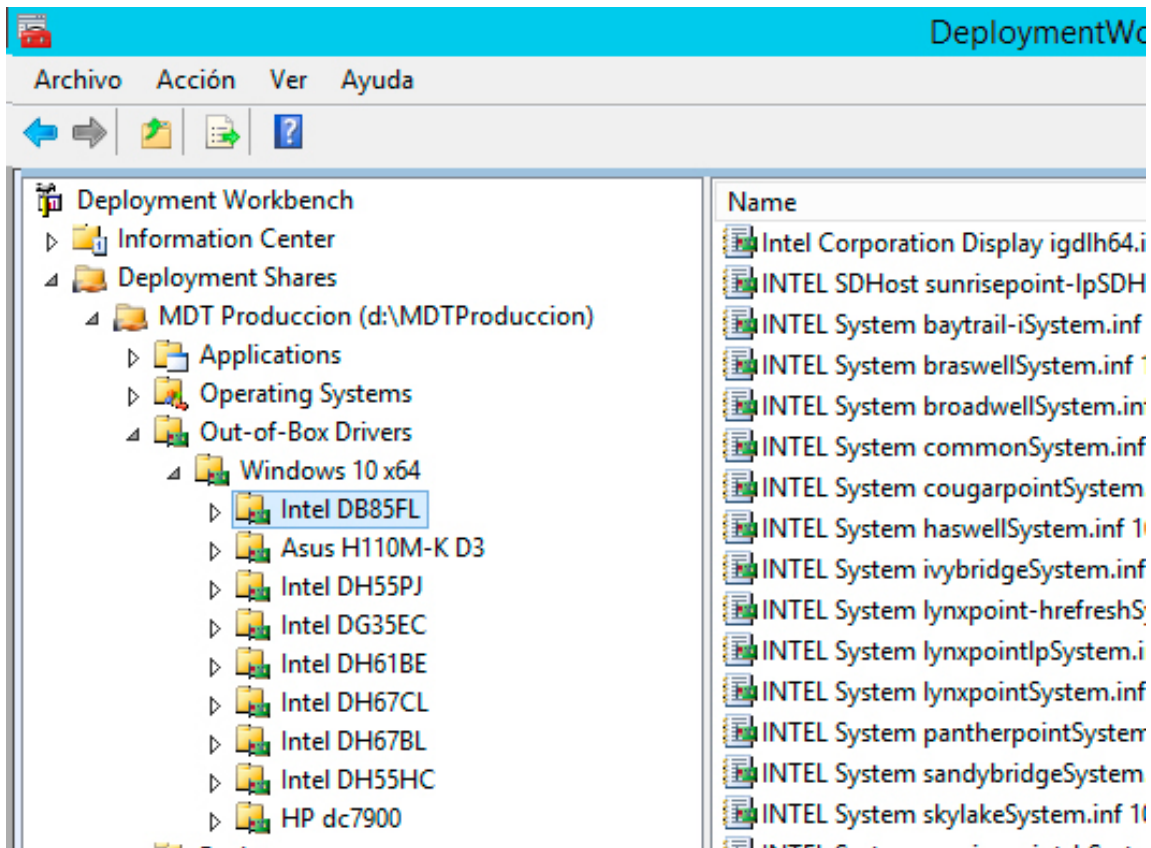


Figura A8: Estructura de carpetas actual para los controladores de Windows 10.

Para que funcione la inyección de controladores tenemos que asignar una variable en la secuencia de tareas que utilice el nuevo nombre amigable. Hay que asignar a la variable DriverGroup001 a Windows 10 x64\%ModelAlias%, creando un nuevo paso en la secuencia de tareas, Set Task Sequence Variable, justo antes del paso que inyecta los controladores de dispositivo, y realizar la asignación.

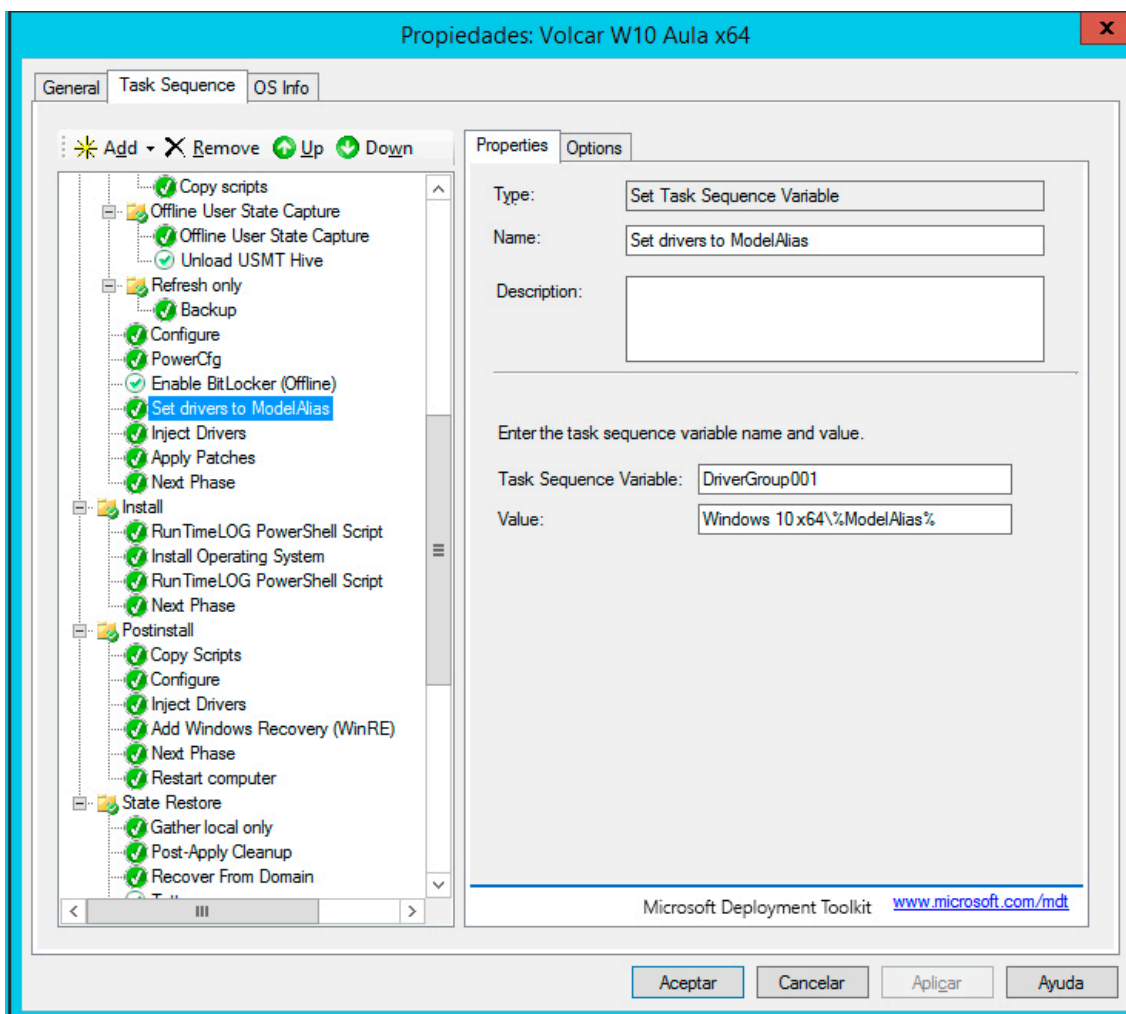


Figura A9: Asignación de la variable DriverGroup001 a Windows 10 x64\;%ModelAlias%.

Como hemos modificado las reglas del DS MDTLab, tenemos que actualizar el DS, de acuerdo al punto 4.4.1.2., se generarán nuevas imágenes de arranque que utilizarán el nuevo *script*. Esta solución es la mejor que he podido encontrar, dado que si Microsoft soluciona el fallo, sin más que editar el *script* externo *AliasUserExit.vbs* y adecuarlo a las respuestas de los comandos WMI sobre el nombre del fabricante (la marca y el modelo), podemos adaptar el *script* rápidamente; y se puede mantener la misma estructura de carpetas para los controladores de dispositivos.